



TUGAS AKHIR - TE 141599

*RANCANG BANGUN SISTEM PENGAMAN TRANSMISI VIDEO PADA
PESAWAT TANPA AWAK*

Fery Setiawan
NRP 07111645000059

Dosen Pembimbing
Ronny Mardiyanto, ST., MT., Ph.D.
Astria Nur Irfansyah, ST., M.Eng., Ph.D.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018



TUGAS AKHIR - TE 141599

**RANCANG BANGUN SISTEM PENGAMAN TRANSMISI VIDEO PADA
PESAWAT TANPA AWAK**

Fery Setiawan
NRP 07111645000059

Dosen Pembimbing
Ronny Mardiyanto, ST.,M.T.,Ph.D
Astria Nur Irfansyah, ST., M.Eng.,Ph.D.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT - TE 141599

***VIDEO TRANSMISSION SECURITY SYSTEM PROTOTYPE ON
UNMANNED AERIAL VEHICLE***

Fery Setiawan
NRP 07111645000059

Supervisor
Ronny Mardiyanto, ST., MT., Ph.D.
Astria Nur Irfansyah, ST., M.Eng., Ph.D.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Electrical Technology
Sepuluh Nopember Institute of Technology
Surabaya 2018

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**RANCANG BANGUN SISTEM PENGAMAN TRANSMISI VIDEO PADA PESAWAT TANPA AWAK**” adalah benar-benar hasil karya intelektual sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya orang lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, 20 Juli 2018

Fery Setiawan
NRP. 0711164500059

.....*Halaman ini sengaja dikosongkan*.....

**RANCANG BANGUN SISTEM PENGAMAN TRANSMISI
VIDEO PADA PESAWAT TANPA AWAK**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik**

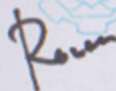
Pada

**Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro**

Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing I,



Dosen Pembimbing II,



Ronny Mardiyanto, S.T., M.T., Ph.D.

NIP: 198101182003121003

Astria Nur Irfansyah, S.T., M.Eng., Ph.D.

NIP: 198103252010121002



.....*Halaman ini sengaja dikosongkan*.....

RANCANG BANGUN SISTEM PENGAMAN TRANSMISI VIDEO PADA PESAWAT TANPA AWAK

Nama : Fery Setiawan
Pembimbing I : Ronny Mardiyanto, ST., MT., Ph.D.
Pembimbing II : Astria Nur Irfansyah, S.T., M.Eng., Ph.D.

ABSTRAK

Sistem pengaman transmisi video adalah sebuah sistem yang mampu mengamankan transmisi dengan cara mengacak gambar sebelum ditransmisikan. Sistem pengaman ini kemudian diaplikasikan pada pesawat tanpa awak atau *Unmanned Aerial Vehicle (UAV)*. Pada UAV, sistem pengaman transmisi diperlukan karena selama ini data yang dikirimkan UAV bersifat terbuka dan dapat diakses oleh semua orang. Gambar dari kamera yang dipasang pada UAV diproses oleh perangkat lunak pada raspberry dengan cara dipotong menjadi beberapa bagian sesuai kebutuhan pengguna, kemudian potongan-potongan tersebut dirangkai secara acak menggunakan algoritma *Linear Feedback Shift Register (LFSR)*. LFSR menggunakan sebuah seed yang bertindak menyerupai kunci untuk mengatur pola pengacakan. Kemudian gambar yang telah diacak tersebut dikirim menggunakan *Transmitter Radio* dengan frekuensi pembawa 5,8 GHz. Karena gambar yang teracak harus dapat dibaca kembali dengan benar oleh pengguna maka dari itu perlu dilakukan rekonstruksi atau biasa disebut dekripsi. Transmisi video yang diterima oleh pengguna pada ground station diubah menjadi standar USB oleh Video Grabber untuk kemudian diproses oleh Raspberry.

Beberapa pengujian dan analisis telah dilakukan untuk memeriksa kualitas gambar hasil pengacakan dan juga gambar hasil dekripsi serta dilakukan pengujian latensi untuk mengukur lama proses penerimaan gambar dan proses dekripsi. Hasil pengujian menunjukkan bahwa tingkat similaritas antara gambar asli dan yang teracak adalah sebesar 3,94 persen. Sementara similaritas antara gambar asli dengan hasil dekripsi atau rekonstruksi yang dilakukan sebesar 98,24 persen. Hasil uji latensi menunjukkan bahwa lama proses penerimaan gambar dan dekripsi adalah 2,2 detik.

Kata kunci: *Unmanned Aerial Vehicle*, Raspberry, dekripsi, *Linear Feedback Shift Register*

.....*Halaman ini sengaja dikosongkan*.....

VIDEO TRANSMISSION SECURITY SYSTEM PROTOTYPE ON UNMANNED AERIAL VEHICLE

Name : Fery Setiawan
Supervisor 1 : Ronny Mardiyanto, ST., MT., Ph.D.
Supervisor II : Astria Nur Irfansyah, S.T., M.Eng., Ph.D.

ABSTRACT

Video transmission security system is a system that capable to secure transmission by randomizing the image before its transmitted. This security system then applied to Unmanned Aerial Vehicle (UAV). In the UAV, the transmission system is required because all the data that UAV sends is open and accessible to everyone. Images from cameras mounted on UAVs are processed by software on Raspberry by cutting into sections according to user requirements, then the pieces are randomly arranged using the Linear Feedback Shift Register (LFSR) algorithm. LFSR uses a seed that acts like a key to set the pattern of randomization. Then the randomized image is sent using Radio Transmitter with 5.8 GHz carrier frequency. Because the random image must be read properly by the user then it needs to be done the reconstruction or commonly called decryption. Transmission of video received by the user on the ground station converted into a USB standard by Video Grabber to then be processed by Raspberry.

Several tests and analyzes have been conducted to check the quality of the randomized images as well as the decrypted images and to test the latency to measure the length of the process of receiving the image and the decryption process. The test results show that the similarity level between the original and the scrambled images is 3.94 percent. While the similarity between the original image with the results of decryption or reconstruction performed for 98.24 percent. Latency test results show that the length of the process of receiving images and decryption is 2.2 seconds.

Keywords: Unmanned Aerial Vehicle, encryption, decryption, pseudo random, Linear Feedback Shift Register

.....*Halaman ini sengaja dikosongkan*.....

KATA PENGANTAR

Alhamdulillah, syukur yang tiada henti penulis panjatkan kehadirat Allah SWT serta tidak lupa sholawat serta salam semoga tetap tercurah kepada junjungan kita Nabi Muhammad SAW sehingga penelitian dalam tugas akhir ini dapat berjalan lancar dan selesai tepat pada waktunya.

Selama pelaksanaan penelitian Tugas Akhir ini, penulis mendapatkan bantuan dari berbagai pihak, dan penulis sampaikan rasa terima kasih. Terima kasih yang sebesar-besarnya juga penulis sampaikan kepada berbagai pihak yang mendukung dan membantu dalam tugas akhir ini, diantaranya:

1. Kedua orang tua yang senantiasa memberikan do'a, nasihat, motivasi, dukungan dan karena keberadaan merekalah penulis tetap semangat untuk menyelesaikan penelitian ini.
2. Ronny Mardiyanto, ST., MT., Ph.D. dan Astria Nur Irfansyah, S.T., M.Eng., Ph.D. selaku dosen pembimbing, atas dukungan moriil maupun materiil, serta bimbingan, inspirasi, pengarahan, dan motivasi yang diberikan selama pengerjaan penelitian tugas akhir ini.
3. Seluruh dosen-dosen Departemen Teknik Elektro, khususnya dosen-dosen bidang studi Elektronika.
4. Rekan-rekan yang banyak membantu dalam penyelesaian tugas akhir ini.

Penulis sadar bahwa Tugas Akhir ini masih belum sempurna dan masih banyak hal yang perlu diperbaiki. Saran, kritik dan masukan baik dari semua pihak sangat membantu penulis terutama untuk berbagai kemungkinan pengembangan lebih lanjut.

Terakhir, penulis berharap Tugas Akhir ini dapat memberikan manfaat bagi banyak pihak dan dapat membantu pengembangan tentang Unmanned Aerial Vehicle kedepannya.

Surabaya, Juni 2018

Fery Setiawan

.....*Halaman ini sengaja dikosongkan*.....

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN KEASLIAN	iii
LEMBAR PENGESAHAN	v
ABSTRAK	vii
<i>ABSTRACT</i>	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah	3
1.3. Batasan Masalah	3
1.4. Tujuan Penelitian	3
1.5. Metodologi Penelitian	3
1.6. Sistematika Penulisan	5
1.7. Relevansi	6
BAB II TINJAUAN PUSTAKA	7
2.1. Kriptografi	7
2.2. Enkripsi Televisi	8
2.3. Pseudo Random Generator	10
2.4. Linear Feedback Shift Register	10
2.5. First Person View (FPV)	11
2.6. Transmitter TS 5828	12
2.7. Receiver Quantum Diversity 5.8GHz	13
2.8. Video Grabber	14
2.9. Unmanned Aerial Vehicle (UAV)	15
2.10. USB Webcam	16
2.11. Raspberry Pi 3	16
2.12. Bahasa Pemrograman C++	17
2.13. Visual Studio	18
2.14. Kate Text Editor	19
2.15. GNU C Compiler	19
2.16. OpenCV	21
2.17. Region of Interest	21

2.18.	Matlab.....	22
2.19.	Koefisien Korelasi	23
2.20.	Composite Video	23
2.21.	Buck Converter.....	24
2.22.	Baterai Lithium Polymer (LiPo).....	25
BAB III PERANCANGAN SISTEM		27
3.1.	Spesifikasi Sistem.....	28
3.2.	Identifikasi Kebutuhan.....	28
3.3.	Perancangan Mekanik.....	29
3.4.	Perancangan Perangkat Keras.....	30
3.5.	Persiapan Raspberry Pi 3	30
3.6.	Perancangan Pemancar	32
3.7.	Perancangan Pengaman Transmisi.....	33
3.8.	Penempatan pada Unmanned.....	41
3.9.	Langkah Penggunaan Alat	42
BAB IV PENGUJIAN DAN ANALISIS		45
4.1.	Pengujian Hardware.....	45
4.2.	Pengujian Software (Enkripsi).....	46
4.3.	Pengujian Software Dekripsi	51
4.4.	Pengujian Software Dekripsi	53
4.5.	Pengujian Kemiripan Gambar	56
4.6.	Pengujian pada UAV.....	57
4.7.	Pengujian Waktu Tunda (Latensi).....	58
BAB V PENUTUP		61
5.1.	Kesimpulan.....	61
5.2.	Saran	61
DAFTAR PUSTAKA		63
LAMPIRAN A.....		65
LAMPIRAN B.....		69
LAMPIRAN G		73
BIODATA PENULIS		75

DAFTAR GAMBAR

Gambar 1.1. Pasukan AS menguji UAV di Chu Lai, Vietnam [1].....	2
Gambar 2.1. Scrambling Chanel TV Analog [2].....	9
Gambar 2.2. Representasi LFSR [3]	11
Gambar 2.3. Goggle FPV [5]	12
Gambar 2.4. TS5828 [6].....	13
Gambar 2.5. Qunam Diversity	14
Gambar 2.7. Bentuk Fisik UAV [9].....	15
Gambar 2.8. Bentuk Fisik Rasperry Pi 3 [10]	17
Gambar 2.8. Board Buck Converter [13]	25
Gambar 2.9. Baterai Lithium Polymer [14]	25
Gambar 3.1. Skema Sistem Keseluruhan	27
Gambar 3.3. Diagram Blok Perangkat Keras	30
Gambar 3.4. Diagram Pengkabelan Transmitter	32
Gambar 3.5. Diagram Pengkabelan Receiver	33
Gambar 3.6. Diagram Blok Pengamanan Transmisi	34
Gambar 3.7. Diagram Blok Proses Enkripsi	34
Gambar 3.8. Ilustrasi penukaran gambar	36
Gambar 3.9. Diagram Blok Proses Dekripsi	38
Gambar 3.10. Pemasangan sistem pada UAV	42
Gambar 4.1. Gambar Asli	47
Gambar 4.2. Enkripsi seed 44257	47
Gambar 4.3. Enkripsi seed 1234	47
Gambar 4.4. Enkripsi seed 1111	47
Gambar 4.5. Gambar Asli	48
Gambar 4.6. Enkripsi seed 44257	48
Gambar 4.7. Enkripsi seed 1234	48
Gambar 4.8. Enkripsi seed 1111	48
Gambar 4.9. Gambar Asli	48
Gambar 4.10. Enkripsi seed 44257	48
Gambar 4.11. Enkripsi seed 1234	48
Gambar 4.12. Enkripsi seed 1111	48
Gambar 4.13. Diagram Blok Enkripsi.....	50
Gambar 4.14. Gambar yang tertangkap kamera.....	50
Gambar 4.15. Hasil Enkripsi Gambar dari kamera	51
Gambar 4.16. Gambar Enkripsi	52

Gambar 4.17. Dekripsi seed 44257.....	52
Gambar 4.18. Dekripsi seed 1234.....	52
Gambar 4.19. Dekripsi seed 1111	52
Gambar 4.20. Gambar Enkripsi	53
Gambar 4.21. Dekripsi seed 44257.....	53
Gambar 4.22. Dekripsi seed 1234.....	53
Gambar 4.23. Dekripsi seed 1111	53
Gambar 4.24. Gambar tangkapan WebCam.....	54
Gambar 4.25. Gambar yang diterima Receiver.....	54
Gambar 4.26. Gambar hasil Crop	55
Gambar 4.27. Gambar hasil Dekripsi	55
Gambar 4.28. Screenshoot MatLab	56
Gambar 4.29 Sinyal gambar yang diterima	57
Gambar 4.30. Sinyal gambar yang terenkripsi.....	58
Gambar 4.31. Pengesetan stopwatch	58
Gambar 4.32. Selisih stopwatch	59

DAFTAR TABEL

Tabel 2.1. Spesifikasi TS 5828 [6]	13
Tabel 4.1 Hasil Pengukuran Daya Sistem	45
Tabel 4.2. Pengujian Transmisi Video	46
Tabel 4.2. Similaritas Gambar pada Kasus 1	49
Tabel 4.3. Similaritas Gambar pada Kasus 2	49
Tabel 4.4. Similaritas Gambar pada Kasus 3	49

.....*Halaman ini sengaja dikosongkan*.....

BAB I

PENDAHULUAN

Tugas Akhir adalah suatu penelitian yang bersifat mandiri yang dilakukan sebagai persyaratan akademik untuk mendapatkan gelar sarjana teknik di Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Topik yang dibahas dalam Tugas Akhir ini adalah pengaman transmisi video pada pesawat tanpa awak.

Pada Bab ini membahas mengenai hal-hal yang mendahului pelaksanaan tugas akhir. Hal tersebut meliputi: latar belakang, perumusan masalah, batasan masalah, tujuan, metodologi, sistematika penulisan dan relevansi

1.1. Latar Belakang

Penggunaan dan pengembangan teknologi pesawat tanpa awak atau *Unmanned Aerial Vehicle (UAV)* muncul sejak awal abad 19, sebelum perang dunia I, pertama kali di tanggal 22 Agustus 1849. Pada saat itu, terjadi pertempuran antara Austria melawan kota Venesia, Italia. Austria yang menguasai mayoritas wilayah Italia dan meluncurkan ratusan balon dari kapal Austria Vulcano. Balon-balon tersebut berhasil mengenai target, walaupun beberapa diantaranya justru meleset berubah arah karena tertiup angin dan malah meledak di perbatasan Austria dan Italia.

Pada tanggal 8 November 1898, Nicolas Tesla, penemu berkebangsaan Amerika Serikat keturunan Serbia mematenkan *remote control* atau pengendali jarak jauh. *Remote control* ini menjadi dasar ilmu robotik kontemporer. Tesla membuat kapal dan balon yang bisa dikendalikan dari jarak jauh. Kemudian pada era Perang Dunia II, UAV digunakan untuk alat latihan para tentara untuk menembak target. Militer Jerman dari pemerintahan Nazi juga menggunakan UAV sebagai senjata udara sepanjang Perang Dunia II.

Kelahiran UAV Amerika dimulai pada 1959, ketika Angkatan Udara AS, khawatir kehilangan pilot di atas wilayah musuh dan mulai merencanakan penerbangan tanpa awak. Setelah Soviet berhasil menembak pesawat mata-mata mereka U-2 pada tahun 1960, program UAV yang sangat rahasia diluncurkan dengan kode "*Red Wagon*". UAV era modern digunakan pertama selama 2 Agustus dan 4 Agustus 1964 ketika terjadi bentrokan di Teluk Tonkin antara AS dan Vietnam Utara.



Gambar 1.1. Pasukan AS menguji UAV di Chu Lai, Vietnam [1]

Ketika Perang Yom Kipur 1973, Israel mengembangkan UAV pertama dengan *real-time surveilans*. Setelah itu rudal permukaan udara Soviet yang digunakan Mesir dan Suriah bisa digempur jet Israel hingga rusak parah. Gambar dan radar *decoying* disediakan oleh UAV ini membantu Israel untuk menetralsir pertahanan udara Suriah pada awal 1982, ketika Perang Libanon, sehingga tidak ada pilot yang tewas. Pada tahun 1987, Israel telah mengembangkan UAV berbasis siluman, Perkembangan teknologi UAV tumbuh pesat selama tahun 1980 dan 1990 yang digunakan selama Perang Teluk Persia pada 1991 dan menjadi mesin pertempuran lebih murah dan lebih mampu.

Perkembangan teknologi pesawat tanpa awak sangatlah pesat. Penggunaan sistem yang pada awalnya hanya digunakan pada militer yang bertujuan sebagai pengintai maupun sebagai peluru kendali. Namun seiring berjalannya waktu sistem ini dikembangkan untuk kebutuhan non-militer maupun komersial seperti pemadam api, pemeriksaan jalur pipa, fotografi maupun sebagai pengirim barang. *Unmanned Aerial Vehicle* banyak mendapat perhatian dari berbagai kalangan karena dapat menggantikan peran manusia sebagai sistem pengendali.

Salah satu keunggulan dari pesawat tanpa awak adalah mampu mengirimkan data transmisi kepada pengguna dengan jarak tertentu. Salah satu data transmisi yang sangat sering dipakai adalah transmisi video. Sistem *PAL (Phase Alternating Line)* yang digunakan sebagai transmisi sangat mudah untuk diakses oleh orang lain, padahal untuk

beberapa kasus, video transmisi bersifat rahasia dan hanya orang-orang tertentu yang boleh mengaksesnya. Untuk mengatasi hal ini, dibutuhkan sebuah sistem keamanan transmisi video yang hanya dapat diakses dengan benar oleh orang-orang tertentu.

Oleh karena itu, penelitian ini dilakukan untuk merancang dan membangun sebuah sistem pengaman transmisi video sederhana dengan mengaplikasikan algoritma *Linear Feedback Shift Register* dan *OpenCV* sebagai library visual

1.2. Perumusan Masalah

Permasalahan yang dibahas dalam tugas akhir ini adalah:

1. Bagaimana mengamankan transmisi video
2. Bagaimana mengaplikasikan sistem pada pesawat tanpa awak

1.3. Batasan Masalah

Batasan masalah dari tugas akhir ini adalah sebagai berikut:

1. Sistem tidak tahan air
2. Penggunaan *OpenCV* sebagai library visual
3. Resolusi video dekripsi diperkecil.
4. Tidak mencakup pembuatan Pesawat Tanpa Awak (*UAV*)

1.4. Tujuan Penelitian

Penelitian pada tugas akhir ini bertujuan sebagai untuk mengembangkan teknologi pesawat tanpa awak dengan cara meningkatkan keamanan transmisi video menggunakan pengacakan gambar sebelum ditransmisikan

1.5. Metodologi Penelitian

Langkah-langkah yang dikerjakan pada tugas akhir ini adalah sebagai berikut:

1. Studi literatur

Studi literatur berisi serangkaian kegiatan pengumpulan dan pengkajian dasar teori yang relevan dan terpercaya untuk menunjang penulisan tugas akhir ini. Literatur dapat bersumber dari paper, jurnal, artikel, buku, maupun *website*, yang bertaraf nasional dan internasional, serta dari hasil konsultasi dengan dosen pembimbing.

Namun, terdapat satu sumber rujukan utama yang telah dilakukan yakni *IEEE Xplore digital library*.

2. Observasi dan Analisa Masalah

Pada tahap ini dilakukan pengkajian terhadap sistem-sistem transmisi video yang sudah ada, meneliti bagaimana membuat enkripsi secara sederhana, serta mengaplikasikannya dalam program pada Raspberry.

3. Persiapan Alat dan Bahan

Persiapan dilakukan agar dalam perancangan alat menjadi lebih terstruktur. Tahap ini merupakan tahap pencarian informasi mengenai konsep-konsep yang dibutuhkan untuk merancang sistem pengaman video ini, yang didapatkan dari studi literatur disertai dengan bimbingan yang intensif dengan dosen. Kemudian dilakukan pengumpulan alat dan bahan yang dibutuhkan.

4. Perancangan Perangkat Keras (*Hardware*)

Perancangan bertujuan mendapatkan desain dan mekanisme yang optimal dengan memperhatikan data-data sebelumnya. *Hardware* yang digunakan meliputi Raspberry Pi 3, dan sebuah EasyCap (*Composite Video Capture*).

5. Perancangan Perangkat Lunak (*Software*)

Pada tahap ini dilakukan perancangan pengembangan pemrograman. Pemrograman meliputi enkripsi data video yang akan dipancarkan (ditransmisikan) pada raspberry. Dan program untuk mendekripsi data video yang dipancarkan pada raspberry yang lain.

6. Tahap Pengujian

Pengujian dilakukan bertahap. Pertama dilakukan pengujian pada pada sistem enkripsi. Sistem diharapkan mampu menampilkan video yang tidak dapat dikenali oleh mata. Kemudian data ditransmisikan ke Raspberry yang lain dengan menggunakan *VideoGrabber*

Tahap pengujian selanjutnya dilakukan dengan mendeskripsikan data yang diterima. Proses dekripsi diawali dengan proses *cropping*, *mapping* dan kemudian *swapping*.

Pengujian berikutnya dilakukan dengan mengubah *seed*

enkripsi (sebuah angka mirip password) dan menguji apakah *seed* yang berbeda akan menghasilkan hasil enkripsi maupun dekripsi yang berbeda.

7. Analisa dan Evaluasi

Analisa dilakukan terhadap hasil pengujian sehingga karakteristik *software* dan *hardware* dapat diketahui. Analisa dilakukan pada dua bagian utama, yaitu pada sisi enkripsi dan sisi dekripsi

8. Penyusunan Laporan Tugas Akhir

Tahap penyusunan laporan merupakan tahap terakhir dari proses pengerjaan tugas akhir ini. Laporan berisi seluruh hal yang berkaitan dengan tugas akhir yang telah dikerjakan meliputi pendahuluan, studi literatur, tinjauan pustaka, perancangan dan pembuatan sistem, pengujian dan analisa, serta penutup.

1.6. Sistematika Penulisan

Dalam buku tugas akhir ini, pembahasan mengenai sistem yang dibuat terbagi menjadi lima bab dengan sistematika penulisan sebagai berikut:

- **BAB I : PENDAHULUAN**

Bab ini meliputi penjelasan latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, sistematika penulisan, dan relevansi.

- **BAB II : TINJAUAN PUSTAKA**

Bab ini menjelaskan tentang teori penunjang yang terkait maupun yang dibutuhkan dalam pengerjaan tugas akhir ini. Dasar teori yang menunjang meliputi *Unmanned Aeroplane Vehicle (UAV)*, Raspberry Pi3, *Webcam Camera*, *EasyCap*, bahasa pemrograman C, *Pseudo Random*, algoritma *Linear Feedback Register*, dan *OpenCV*. Selain itu, bagian ini juga berisi tinjauan pustaka untuk membandingkan sistem yang telah ada dengan yang dirancang pada tugas akhir ini.

- **BAB III : PERANCANGAN SISTEM**

Bab ini menjelaskan tentang perencanaan sistem yang meliputi perangkat keras (*hardware*) dan perangkat lunak (*software*) untuk Pembuatan Rancang Bangun Sistem Pengaman Transmisi Video

pada Pesawat Tanpa Awak.

- **BAB IV : PENGUJIAN DAN ANALISIS**

Bab ini menjelaskan tentang hasil uji coba alat beserta analisisnya.

- **BAB V : PENUTUP**

Bab ini berisi tentang kesimpulan yang diperoleh dari pembuatan alat serta saran untuk pengembangan lebih lanjut.

1.7. Relevansi

Sistem Pengaman Transmisi Video merupakan sebuah inovasi untuk memberikan keamanan tambahan pada video yang ditransmisikan terutama pada pesawat tanpa awak.

BAB II

TINJAUAN PUSTAKA

Suatu penelitian memerlukan teori-teori yang sudah ada sebelumnya untuk dikaji lebih dalam demi memperkuat argumen penulis. Teori tersebut digunakan untuk membantu penulis dan sebagai dasar dalam membuat suatu penelitian.

Pada bab ini terdapat beberapa teori dasar yang menjadi landasan untuk merumuskan dan menyelesaikan masalah yang akan dibahas pada penelitian ini. Pada bagian awal terdapat tinjauan pustaka mengenai pengamanan video yang meliputi kriptografi, *Pseudo Random Number Generator*, *Linear Feedback Shift Register* dan Segmentasi Gambar. Kemudian dibahas pula mengenai teknik Transmisi Video yang meliputi *First Person View*, *Transmitter* dan *Receiver*. Yang terakhir adalah teori-teori penunjang perangkat keras dan perangkat lunak

2.1. Kriptografi

Kriptografi (*cryptography*) berasal dari bahasa Yunani, terdiri dari dua suku kata yaitu *crypto* dan *graphy*. *Crypto* artinya menyembunyikan, sedangkan *graphy* artinya tulisan. Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data. Tetapi tidak semua aspek keamanan informasi dapat diselesaikan dengan kriptografi.

Kriptografi dapat pula diartikan sebagai ilmu atau seni untuk menjaga keamanan pesan. Pada prinsipnya, Kriptografi memiliki 4 komponen utama yaitu:

1. *Plaintext*, yaitu pesan yang dapat dibaca
2. *Ciphertext*, yaitu pesan acak yang tidak dapat dibaca
3. *Key*, yaitu kunci untuk melakukan teknik kriptografi
4. Algoritma, yaitu metode untuk melakukan enkripsi dan dekripsi

Kriptografi modern sangat didasarkan pada teori matematika dan praktik ilmu komputer. Algoritma kriptografi dirancang berdasarkan asumsi-asumsi komputasional. Membuat algoritma-algoritma semacam itu sulit dipecahkan. Secara teoritis mungkin untuk mematahkan sistem semacam itu, tetapi tidak dapat dilakukan oleh sarana praktis yang dikenal. Karenanya skema ini disebut aman secara komputasi.

Kemajuan teoritis, misalnya peningkatan dalam algoritma faktorisasi bilangan bulat, dan teknologi komputasi yang lebih cepat

memerlukan solusi ini untuk terus disesuaikan. Terdapat skema keamanan teoritis tentang informasi yang mungkin tidak dapat diputuskan bahkan dengan daya komputasi yang tidak terbatas

Pertumbuhan teknologi kriptografi telah mengangkat sejumlah masalah hukum di era informasi. Potensi kriptografi untuk digunakan sebagai alat untuk spionase dan hasutan telah menyebabkan banyak pemerintah mengklasifikasikannya sebagai senjata dan membatasi atau bahkan melarang penggunaan dan ekspornya. Di beberapa yurisdiksi dimana penggunaan kriptografi adalah legal, undang-undang mengizinkan penyelidikan untuk memaksa pengungkapan kunci enkripsi untuk dokumen yang relevan dengan penyelidikan. Kriptografi juga memainkan peran utama dalam manajemen hak digital dan pelanggaran hak cipta dari media digital.

2.2. Enkripsi Televisi

Enkripsi televisi, sering disebut sebagai "*scrambling*", adalah enkripsi yang digunakan untuk mengontrol akses ke layanan televisi berbayar, biasanya layanan televisi kabel atau satelit. Televisi prabayar menghasilkan pendapatan dari pelanggan, dan terkadang pelanggan tersebut tidak membayar. Pencegahan pembajakan pada jaringan kabel dan satelit telah menjadi salah satu faktor utama dalam pengembangan sistem enkripsi TV berbayar.

Teknik lain seperti menambahkan sinyal pengganggu ke video atau audio mulai digunakan sebagai solusi sederhana. Seiring berkembangnya teknologi, teknik pengacakan yang lebih rumit seperti enkripsi digital dari pemotongan atau pemutaran audio dan video, dimana garis video dipotong pada titik tertentu dan kedua bagian tersebut kemudian diatur ulang dan diterapkan pada sinyal.

Enkripsi digunakan untuk melindungi *feed* yang didistribusikan melalui satelit untuk jaringan televisi kabel. Di Perancis, *Chanel* + meluncurkan layanan *scrambling* pada tahun 1984. *Chanel* + juga mengklaim bahwa itu adalah sistem yang tidak dapat dipecahkan. Sayangnya, sebuah majalah elektronik, "*Radio Plans*", menerbitkan desain untuk *decoder* dalam waktu satu bulan sejak peluncuran saluran.

Di Amerika Serikat, HBO adalah salah satu layanan pertama yang mengenkripsi sinyal menggunakan sistem *VideoCipher* II. Di Eropa, FilmNet mengacak layanan satelitnya pada bulan September 1986, sehingga menciptakan salah satu pasar terbesar untuk dekoder TV satelit di dunia, karena sistem yang digunakan FilmNet dengan mudah diretas.

Televisi berbayar analog dan digital memiliki beberapa sistem akses bersyarat yang digunakan untuk bayar-per-tayang (*PPV*) dan layanan terkait pelanggan lainnya. Awalnya, sistem televisi kabel analog hanya mengandalkan *set-top box* untuk mengontrol akses ke pemrograman, karena televisi awalnya tidak "siap kabel". Enkripsi analog biasanya terbatas pada saluran premium seperti HBO atau saluran dengan konten berorientasi dewasa. Dalam kasus-kasus tersebut, berbagai metode penekanan sinkronisasi video eksklusif digunakan untuk mengontrol akses ke pemrograman. Pada beberapa sistem ini, sinyal sinkronisasi yang diperlukan berada pada *subcarrier* terpisah meskipun terkadang polaritas sinkron hanya terbalik, dalam hal ini, jika digunakan bersama dengan *PAL*, *TV SECAM L* dengan *tuner* kabel dapat digunakan untuk menguraikan sinyal meskipun hanya dalam hitam dan putih dan dengan pencahayaan terbalik dan dengan demikian TV multi standar yang mendukung *PAL L* lebih disukai untuk memecahkan kode warna juga. Namun, ini akan menyebabkan bagian dari sinyal video diterima sebagai audio juga dan dengan demikian TV lain dengan sebaiknya tidak ada *auto mute* harus digunakan untuk audio decoding. Kotak *set-top* analog sebagian besar telah digantikan oleh kotak *set-top* digital yang dapat secara langsung mengontrol akses ke pemrograman serta sinyal dekripsi secara digital.

VideoCipher II RS (VCII RS) adalah sistem pengacakan yang digunakan saluran TV satelit C-Band. Penerima satelit yang mampu VCII diperlukan untuk memecahkan kode saluran VCII. VCII sebagian besar telah digantikan oleh *DigiCipher 2* di Amerika Utara. Awalnya, receiver berbasis VCII memiliki teknologi modem terpisah untuk akses bayar per tayang yang dikenal sebagai *Videopal*. Teknologi ini menjadi terintegrasi sepenuhnya dalam *receiver* televisi satelit analog generasi selanjutnya.



Gambar 2.1. *Scrambling Chanel TV Analog [2]*

2.3. *Pseudo Random Generator*

Pseudo random adalah proses pengacakan yang berpola. Jadi sebenarnya pengacakan yang dilakukan tidak benar-benar acak. Urutan *pseudorandom* biasanya menunjukkan keacakan secara statistik ketika sedang dihasilkan proses kausal yang sepenuhnya *deterministic*. Proses seperti ini lebih mudah diproduksi daripada yang benar-benar acak, dan memiliki manfaat yang dapat digunakan lagi dan lagi untuk menghasilkan angka yang sama persis, yang berguna untuk menguji perangkat lunak.

Pembangkitan angka secara acak memiliki beberapa kegunaan, mayoritas pada statistik, untuk sampling acak dan simulasi. Sebelum peneliti komputer modern menemukan *pseudorandom*, mereka menggunakan dadu, kartu atau roda rolet (semacam roda yang diputar dan akan berhenti di suatu nilai tertentu) untuk mendapatkan urutan angka secara acak.

Penemuan pertama metode untuk menghasilkan sebuah digit yang acak adalah pada tahun 1927, ketika Universitas Cambridge menerbitkan 41.600 digit tabel acak yang dikembangkan L.H.C. Tippett [4]

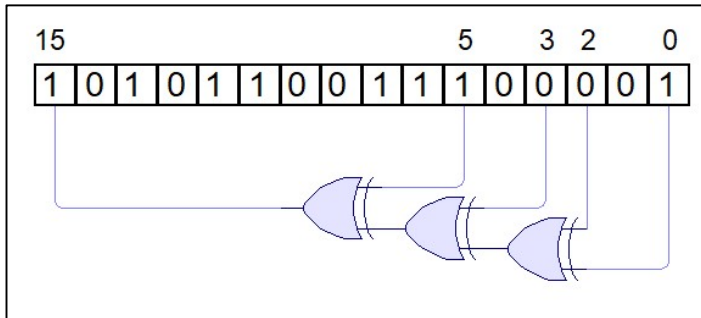
2.4. *Linear Feedback Shift Register*

Linear Feedback Shift Register (LFSR) adalah register geser yang bit inputnya merupakan fungsi linear dari keadaan sebelumnya. Fungsi linear yang paling umum digunakan dari bit tunggal adalah eksklusif-OR (*XOR*). Jadi, *LFSR* biasanya merupakan register geser yang bit inputnya didorong oleh *XOR* dari beberapa bit nilai register geser keseluruhan.

Linear Feedback Shift Register digunakan untuk membuat urutan angka yang kemudian digunakan untuk mengacak gambar. Nilai awal *LFSR* disebut benih (*seed*), dan karena pengoperasian register bersifat deterministik, aliran nilai yang dihasilkan oleh register sepenuhnya ditentukan oleh statusnya saat ini (atau sebelumnya). Demikian juga, karena memiliki sejumlah keadaan yang mungkin, akhirnya harus memasukkan siklus yang berulang. Namun, *LFSR* dengan fungsi umpan balik dapat menghasilkan urutan bit yang muncul acak dan memiliki siklus yang sangat panjang.

contoh:

$x^5 + x^3 + x^2 + x^0$ yang merepresentasikan bit umpan balik yang kemudian di *EXOR*-kan. Pin 15 (x^{15}) merepresentasikan *MSB (Most Significant Bit)*, dimana hasil dari gerbang *EXOR* diletakkan. Kemudian bit akan digeser kekanan satu kali. Nilainya merupakan angka berikutnya pada sekuensial.



Gambar 2.2. Representasi *LFSR* [3/]

Linear Feedback function : $x^5 + x^3 + x^2 + x^0$

State saat ini : 1010110011100001

State selanjutnya : 0101011001110000

Sebagai masukan dari *EXOR* bit 0,2,3,5 merupakan logika penentu dari nilai *feedback*. Sebelum nilai *EXOR* dari bit-bit tersebut dijadikan *input feedback*, maka terlebih dahulu register di geser ke kanan sebanyak satu kali.

Aplikasi *LFSR* diantaranya adalah untuk menghasilkan nomor *pseudo-random*, *pseudo-noise sequences*, penghitung digital cepat, dan urutan pemutihan [5].

2.5. First Person View (FPV)

First Person View (FPV), juga dikenal sebagai *Remote Person View (RPV)* adalah metode yang digunakan untuk mengendalikan kendaraan yang dikontrol radio dari pengemudi atau titik pandang pilot. Paling umum digunakan untuk pilot pesawat radio kendali atau jenis lain dari pesawat tanpa awak (*UAV*).

UAV digerakkan dari jarak jauh dari sudut pandang orang pertama melalui kamera onboard, diumpankan secara nirkabel ke video *FPV goggle* atau *monitor video*. Pengaturan yang lebih canggih termasuk *stabilized camera pan* dan *tilt* yang dikendalikan oleh sensor giroskop di kacamata pilot dan dengan kamera *onboard* ganda, memungkinkan pandangan stereoskopik yang sesungguhnya. Ada dua komponen utama dari pengaturan *FPV* yaitu komponen udara dan komponen darat (*ground station*). Sistem *FPV* dasar terdiri dari kamera dan pemancar video analog di pesawat dengan penerima video dan layar pada *ground station*.



Gambar 2.3. Goggle FPV [5]

Setup yang lebih canggih biasanya menambahkan perangkat keras khusus, termasuk tampilan di layar dengan navigasi *GPS* dan data penerbangan, sistem stabilisasi, dan perangkat autopilot dengan kemampuan "kembali ke rumah" yang memungkinkan pesawat terbang kembali ke titik awalnya secara otonom saat kehilangan sinyal. Fitur umum lainnya adalah penambahan kemampuan *pan* dan *tilt* ke kamera, yang disediakan oleh motor servo. Ketika digabungkan dengan kacamata video dan perangkat sensor pada kepala dapat menciptakan pengalaman yang benar-benar imersif, seolah-olah pilot benar-benar duduk di kokpit pesawat *RC*. Peralatan penerima umumnya disebut sebagai "*Ground station*" pada umumnya terdiri dari penerima video analog (pencocokan frekuensi pemancar di pesawat) dan perangkat untuk melihat. *Ground Station* yang lebih kompleks sering menyertakan sarana untuk merekam gambar yang diterima bersama dengan antena yang lebih canggih untuk mencapai jangkauan dan kejelasan yang lebih besar.

2.6. Transmitter TS 5828

TX5828 (ditunjukkan pada Gambar 2.4) adalah sebuah video *transmitter* yang digunakan untuk mentransmisikan data video yang telah dienkripsi. Devais ini bertugas untuk mengirimkan sinyal *PAL/NTSC* melalui Sinyal Radio dengan frekuensi pembawa 5.35-5.95 Ghz. TX 5828 memiliki spesifikasi yang ditunjukkan pada Tabel 2.1



Gambar 2.4. TS5828 [6]

Tabel 2.1. Spesifikasi TS 5828 [6]

No	Keterangan	Spesifikasi
1	Frekuensi Transmisi	ISM 5.35GHz-5.95GHz 48 CH
2	Tegangan Kerja	+6.5V~+25.0V
3	Sumber Arus	250mA±20mA @+12V @600mW
4	Video Input Level	1V±0.2Vp-p/75Ω
5	Audio Input Level	0.5~2.0Vp-p/10K Ω
6	TV Sistem	PAL/NTSC/SECAM (AUTO)
7	Frekuensi pembawa audio	6.5 MHz
8	Daya Keluaran RF	0.25mW(-6dBm±0.5dBm) 25mW(14dBm±0.5dBm) 200mW(23dBm±0.5dBm) 600mW(28dBm±0.5dBm)
9	Konektor Antena	SMA Connector
10	Berat	9 gram
11	Dimensi	31x22x8mm
12	Suhu Kerja	-10°C ~+85

2.7. Receiver *Quantum Diversity* 5.8GHz

Quantum Diversity (ditunjukkan pada Gambar 2.5) adalah 3 *receiver* yang dirangkai bersama-sama, yang berarti dapat diatur ke 3 saluran yang berbeda pada saat yang bersamaan. Alat ini mendeteksi seluruh pita frekuensi 5.8GHz, dan menunjukkan saluran apa yang mentransmisikan sinyal.



Gambar 2.5. *Qunam Diversity*

2.8. *Video Grabber*

Video Grabber adalah sebuah divais yang digunakan untuk mengubah sinyal *Composite* (Standar *PAL/NTSC/SECAM*) agar dapat terbaca oleh standar USB komputer. Di pasaran terdapat beberapa jenis chip. Salah satunya adalah STK1160. *Chip* yang satu ini mempunyai kompatibilitas dengan *Chip* ARM, sedangkan beberapa chip yang lainnya tidak [1].

Spesifikasi *Easycap STK1160* adalah sebagai berikut :

- PAL 25 *frame/det* dengan resolusi 720x576
- NTFS 30 *frame/det* dengan resolusi 720x480
- S-Video input
- CVBS+S-Video model dan CVBS model (4 kanal) [1]



Gambar 2.6. *Video Grabber EasyCap STK1160* [8]

2.9. Unmanned Aerial Vehicle (UAV)

Unmanned Aerial Vehicle (UAV) merupakan sebuah mesin terbang yang dapat dikendalikan jarak jauh maupun bergerak sendiri yang biasanya digunakan untuk mencari informasi, membawa muatan, senjata, dan lain-lain lewat udara. Perkembangan *UAV* hingga saat ini merupakan hasil dari berkembangnya teknologi kontrol otomatis. Kontrol *UAV* ada dua variasi utama, yaitu kontrol melalui pengendali jarak jauh dan kendali otomatis berdasarkan program yang telah dimasukkan ke dalam pesawat. *UAV* sering digunakan dalam misi pengintaian dan penyerangan. Selain itu, *UAV* juga digunakan pada keperluan umum seperti pemadam kebakaran, keamanan, pengecekan, dan banyak lagi[1].

Perkembangan UAV di Indonesia pada 2011 mulai dirintis oleh Lembaga Penerbangan dan Antariksa Nasional (LAPAN). *UAV* yang sedang dikembangkan oleh LAPAN dirancang untuk menjalankan berbagai misi salah satunya ialah misi pemantauan wilayah Indonesia terutama untuk wilayah pegunungan dan pulau-pulau terpencil. Terdapat lima *prototype* UAV yang telah berhasil diciptakan yaitu LSU-01, LSU-02, LSU-03, LSU-04, dan LSU-05.



Gambar 2.7. Bentuk Fisik UAV [9]

2.10. *USB Webcam*

WebCam adalah sebuah *peripheral* berupa kamera sebagai pengambil citra atau gambar yang dikendalikan oleh sebuah komputer atau oleh jaringan komputer. Ada beberapa orang mengartikan *WebCam* sebagai *Web pages + Camera*, karena dengan menggunakan *WebCam* untuk mengambil gambar video secara aktual dapat langsung di upload bila komputer yang mengendalikan terkoneksi internet. *USB Webcam* mendukung fasilitas *plug and play* dan dapat dihubungkan pada port *USB* tanpa harus mematikan komputer.

2.11. *Raspberry Pi 3*

Raspberry Pi 3 yang ditunjukkan pada Gambar 2.8 adalah *single board* mini komputer yang memiliki pin I/O atau *GPIO* yang bekerja menggunakan *operation system* linux. Raspberry menggunakan *System on Chip (SoC)* broadcom BCM2837 dengan CPU ARM cortex-A53 1.2 GHz. Mini komputer ini dibekali dengan RAM sebesar 1 GB dan GPU broadcom *videocore* IV. Peningkatan dari versi sebelumnya raspberry pi 3 telah dilengkapi dengan 10/100 Ethernet, 2.4 GHz 802.11n *wireless*. Untuk penyimpanannya telah menggunakan *microSD* dan 40 pin *GPIO*.

Raspberry Pi, sering disingkat dengan nama Raspi, dapat digunakan untuk menjalankan program perkantoran, permainan komputer, dan sebagai pemutar media hingga video beresolusi tinggi. Raspberry Pi dikembangkan oleh yayasan nirlaba, Raspberry Pi *Foundation*, yang digawangi sejumlah pengembang dan ahli komputer dari Universitas Cambridge, Inggris

Ide dibalik Raspberry Pi diawali dari keinginan untuk mencetak pemrogram generasi baru. Seperti disebutkan dalam situs resmi Raspberry Pi *Foundation*, waktu itu Eben Upton, Rob Mullins, Jack Lang, dan Alan Mycroft, dari laboratorium komputer Universitas Cambridge memiliki kekhawatiran melihat kian turunnya keahlian dan jumlah siswa yang hendak belajar ilmu komputer. Mereka lantas mendirikan yayasan Raspberry Pi bersama dengan Pete Lomas dan David Braben pada 2009. Tiga tahun kemudian, Raspberry Pi model B memasuki produksi massal. Dalam peluncuran pertamanya pada akhir Februari 2012 dalam beberapa jam saja sudah terjual 100.000 unit. Pada bulan Februari 2016, Raspberry Pi *Foundation* mengumumkan bahwa mereka telah menjual 8 juta perangkat Raspi, sehingga menjadikannya sebagai perangkat paling laris di Inggris.



Gambar 2.8. Bentuk Fisik Raspberry Pi 3 [10]

Raspberry Pi memiliki dua model: model A dan model B. Secara umum Raspberry Pi model B memiliki kapasitas penyimpanan RAM sebesar 512 MB. Perbedaan model A dan B terletak pada modul penyimpanan yang digunakan. Model A menggunakan penyimpanan sebesar 256 MB dan penyimpanan model B sebesar 512 MB. Selain itu, model B sudah dilengkapi dengan *port Ethernet* (untuk LAN) yang tidak terdapat di model A. Desain Raspberry Pi didasarkan pada *SoC (system-on-a-chip)* Broadcom BCM2835, yang telah menanamkan prosesor ARM1176JZF-S dengan 700 MHz, GPU VideoCore IV, dan RAM sebesar 256 MB (model B). Penyimpanan data tidak didesain untuk menggunakan kram keras atau *solid-state drive*, melainkan mengandalkan kartu penyimpanan tipe *SD* untuk menjalankan sistem dan sebagai media penyimpanan jangka panjang.

2.12. Bahasa Pemrograman C++

C++ adalah Bahasa pemrograman computer yang ditemukan oleh Bjarne Stroustrup pada tahun 1980an. Bahasa pemrograman ini merupakan pengembangan dari Bahasa C yang telah ditemukan pada awal tahun 1970an oleh Dennis Ritchie yang saat itu bekerja untuk Bell Labs yang berlokasi di Amerika Serikat [2].

Pada perkembangannya, versi ANSI (*American National Standard Institute*) bahasa pemrograman C menjadi versi yang dominan, Meskipun versi tersebut jarang digunakan dalam pengembangan sistem dan jaringan untuk embedded, Bjarne Stroustrup yang bekerja pada Bell Labs pertama kali mengembangkan C++ pada awal 1980-an. Untuk mendukung fitur-fitur pada C++, dibangun dengan efisiensi dan dukungan sistem untuk pemrograman tingkat rendah (*low level coding*).

Pada C++ ditambahkan konsep-konsep baru seperti kelas dengan sifat-sifatnya seperti *inheritance* dan *overloading*. Salah satu perbedaan

yang paling mendasar antara C++ dengan bahasa C adalah bentuk dari konsep-konsep yang berorientasi objek (*Object Oriented Programming*). Pemrograman berorientasi objek (OOP) merupakan paradigma pemrograman berdasarkan konsep "objek", yang dapat berisi data, dalam bentuk field atau dikenal juga sebagai atribut, serta kode dalam bentuk fungsi/prosedur atau dikenal juga sebagai *method*. Semua data dan fungsi di dalam paradigma ini dikemas dalam kelas-kelas atau objek-objek. Dibandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya,

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik peranti lunak skala besar. Lebih jauh lagi, pendukung *OOP* mengklaim bahwa *OOP* lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan *OOP* lebih mudah dikembangkan.

2.13. Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi personal, dan lain-lain. Aplikasinya, dalam bentuk aplikasi *console*, aplikasi Windows, ataupun aplikasi *Web*. *Visual Studio* mencakup kompiler, SDK, *Integrated Development Environment (IDE)*, dan dokumentasi (umumnya berupa MSDN Library).

Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe. Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun *managed code* (dalam bentuk Microsoft Intermediate Language di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan di atas .NET Compact Framework) Android maupun iOS.

Visual Studio kini telah menginjak versi Visual Studio 9.0.21022.08, atau dikenal dengan sebutan Microsoft Visual Studio 2008 yang diluncurkan pada 19 November 2007, yang ditujukan untuk platform Microsoft .NET Framework 3.5. Versi sebelumnya, Visual Studio 2005 ditujukan untuk platform .NET Framework 2.0 dan 3.0. Visual Studio 2003 ditujukan untuk .NET Framework 1.1, dan Visual

Studio 2002 ditujukan untuk .NET Framework 1.0. Versi-versi tersebut di atas kini dikenal dengan sebutan Visual Studio .NET, karena memang membutuhkan Microsoft .NET Framework. Sementara itu, sebelum muncul Visual Studio .NET, terdapat Microsoft Visual Studio 6.0 (VS1998).

2.14. Kate Text Editor

KDE Advanced Text Editor (Kate) adalah editor teks yang dikembangkan oleh komunitas perangkat lunak gratis KDE. *Kate* telah menjadi bagian dari *KDE Software Compilation* sejak rilis 2.2 pada tahun 2001. Karena teknologi *KParts*, memungkinkan untuk menanamkan *Kate* sebagai komponen pengeditan di aplikasi KDE lainnya. Aplikasi KDE utama yang menggunakan *Kate* sebagai komponen pengeditan termasuk lingkungan pengembangan terintegrasi *KDevelop*, lingkungan pengembangan web *Quanta Plus*, dan *LaTeX front-end Kile*.

Kate telah memenangkan perbandingan editor teks tingkat lanjut di *Linux Voice magazine*. Hingga Juli 2014 pengembangan sudah mulai menghubungkan *Kate*, bersama *Dolphin*, *Konsole*, *KDE Telepathy*, dan *Yakuake*, ke *KDE Frameworks 5*. *Kate* adalah editor teks *programmer* yang menampilkan penyorotan sintaks untuk lebih dari 200 format file dengan aturan pelipatan kode. Sorotan sintaks dapat diperluas melalui file XML. Mendukung skema pengkodean UTF-8, UTF-16, ISO-8859-1 dan ASCII dan dapat mendeteksi pengkodean karakter file secara otomatis. *Kate* dapat digunakan sebagai editor teks modal melalui mode input yang mengemulasikan editor teks *Unix* dengan nama yang sama.

Kate memiliki banyak antarmuka dokumen, pemisahan window, pengeditan proyek dan sesi untuk memfasilitasi pengeditan beberapa dokumen. Menggunakan sesi, seseorang dapat menyesuaikan *Kate* untuk berbagai proyek dengan menyimpan daftar file yang terbuka, daftar plugin yang diaktifkan, dan konfigurasi jendela.

Untuk mencari dan mengganti teks, *Kate* mempunyai fitur pencarian tambahan, pencarian *multi-line* dan mengganti teks dan dukungan ekspresi reguler. *Kate* juga dapat melakukan pencarian dan mengganti beberapa file.

2.15. GNU C Compiler

GNU Compiler Collection (GCC) adalah sistem kompiler yang diproduksi oleh GNU project yang mendukung berbagai bahasa pemrograman. *GCC* adalah komponen kunci dari *toolchain* GNU dan compiler standar untuk kebanyakan sistem operasi mirip Unix. *Free*

Software Foundation (FSF) mendistribusikan GCC di bawah *GNU General Public License* (GNU GPL).

GCC telah memainkan peran penting dalam pertumbuhan perangkat lunak bebas, baik sebagai alat dan contoh. Awalnya bernama *GNU C Compiler*, ketika hanya menangani bahasa pemrograman C, GCC 1.0 dirilis pada tahun 1987. Kemudian disempurnakan untuk mengkompilasi C++ pada bulan Desember tahun itu. *Front end* kemudian dikembangkan untuk Objective-C, Objective-C ++, Fortran, Java, Ada, dan Go. Versi 4.5 dari spesifikasi OpenMP sekarang didukung dalam C dan C++ compiler dan implementasi "jauh lebih baik" dari spesifikasi OpenACC 2.0a juga didukung.

Secara default, versi saat ini mendukung gnu ++ 14, superset dari C++ 14, dan gnu11, superset dari C11, dengan dukungan standar yang ketat juga tersedia. Ini juga menyediakan dukungan eksperimental untuk C ++ 17 dan yang lebih baru. GCC telah dipindah ke berbagai arsitektur instruksi set, dan secara luas digunakan sebagai alat dalam pengembangan perangkat lunak bebas dan berpaten. GCC juga tersedia untuk sebagian besar sistem *embedded*, termasuk berbasis ARM; AMCC, dan chip berbasis Arsitektur *Freescape Power*. *Compiler* dapat menargetkan berbagai platform. Selain sebagai compiler resmi sistem operasi GNU, GCC telah diadopsi sebagai compiler standar oleh banyak sistem operasi komputer Unix yang modern, termasuk Linux dan keluarga BSD, meskipun FreeBSD dan macOS telah pindah ke sistem LLVM. Versi juga tersedia untuk Microsoft Windows dan sistem operasi lainnya.

Antarmuka eksternal GCC mengikuti konvensi Unix. Pengguna meminta program driver khusus bahasa (gcc untuk C, g++ untuk C++, dll.), yang menafsirkan argumen perintah, memanggil compiler yang sebenarnya, menjalankan *assembler* pada *output*, dan kemudian secara opsional menjalankan linker untuk menghasilkan biner yang dapat dieksekusi lengkap. Setiap compiler bahasa adalah program terpisah yang membaca kode sumber dan kode mesin keluaran. Semua memiliki struktur internal umum. Sebuah *front end* per-bahasa mem-parsing kode sumber dalam bahasa itu dan menghasilkan sebuah pohon sintaks abstrak ("pohon" untuk pendek).

GCC dapat mengkonversi ke representasi input middle-end, yang disebut *GENERIC form*. *Middle end* kemudian secara bertahap mengubah program menuju bentuk akhirnya. Optimisasi kompilator dan teknik analisis kode statis (seperti FORTIFY_SOURCE, direktif kompilator yang mencoba untuk menemukan beberapa buffer overflows) diterapkan

ke kode. Ini bekerja pada beberapa representasi, kebanyakan representasi GIMPLE, arsitektur-independen dan representasi RTL yang bergantung pada arsitekturnya. Akhirnya, kode mesin diproduksi menggunakan pola pencocokan khusus arsitektur awalnya berdasarkan pada algoritma Jack Davidson dan Chris Fraser.

GCC ditulis terutama di C kecuali untuk bagian dari *front end* Ada. Distribusi mencakup pustaka standar untuk Ada, C++, dan Java yang sebagian besar kode ditulis dalam bahasa tersebut. Pada beberapa platform, distribusi juga menyertakan librari *runtime* tingkat rendah, libgcc, yang ditulis dalam kombinasi kode mesin-independen C dan mesin khusus prosesor, yang dirancang terutama untuk menangani operasi aritmatika yang tidak dapat dilakukan oleh prosesor target secara langsung.

2.16. *OpenCV*

OpenCV (Open Source Computer Vision Library) adalah sebuah pustaka perangkat lunak yang ditujukan untuk pengolahan citra dinamis secara *real-time*, yang dibuat oleh Intel, dan sekarang didukung oleh Willow Garage dan Itseez. Program ini disediakan secara gratis dan dapat dijalankan pada Linux, Windows, dan Mac OS X. Pengembangan juga dilakukan pada Python, Ruby, Matlab dan Bahasa Pemrograman lainnya.. Pustaka ini merupakan pustaka lintas *platform*. Program ini didedikasikan sebageian besar untuk pengolahan citra secara *real-time*. Jika pustaka ini menemukan pustaka *Integrated Performance Primitives* dari Intel dalam sistem komputer, maka program ini akan menggunakan rutin ini untuk mempercepat proses kerja program ini secara otomatis.

OpenCV ditulis dalam bahasa C yang dioptimasi, sehingga dapat menguntungkan dalam segi *multicore processor*. Salah satu tujuan dari diciptakannya *OpenCV* adalah untuk menyediakan *computer infrastructure* yang mudah digunakan yang membantu pengguna untuk membuat suatu aplikasi visual dengan cepat dan mudah. [3]

2.17. *Region of Interest*

Region of Interest (sering disingkat *ROI*), adalah sampel dalam kumpulan data yang diidentifikasi untuk tujuan tertentu. Konsep *ROI* biasanya digunakan di banyak area aplikasi. Sebagai contoh, dalam pencitraan medis, batas-batas tumor dapat didefinisikan pada gambar atau dalam volume, untuk tujuan mengukur ukurannya. Batas endokardial dapat didefinisikan pada gambar, mungkin selama fase yang berbeda dari siklus jantung misalnya, *end-systole* dan *end-diastole*, untuk tujuan

menilai fungsi jantung. Dalam sistem informasi geografis (GIS), ROI dapat diambil secara harfiah sebagai seleksi poligonal dari peta 2D. Dalam visi komputer dan pengenalan karakter optik, ROI mendefinisikan batas objek yang sedang dipertimbangkan. Dalam banyak aplikasi, label simbolis (tekstual) ditambahkan ke ROI, untuk menggambarkan kontennya secara ringkas. Dalam ROI mungkin terdapat *Place of Interest* (POI).

2.18. Matlab

Matlab (*Matrix Laboratory*) adalah sebuah lingkungan komputasi numerikal dan bahasa pemrograman komputer generasi keempat. Dikembangkan oleh The *MathWorks*, MATLAB memungkinkan manipulasi matriks, memplotkan fungsi dan data, implementasi algoritma, pembuatan antarmuka pengguna, dan antarmuka dengan program dalam bahasa lainnya. Meskipun hanya bernuansa numerik, sebuah kotak kakas (*toolbox*) yang menggunakan mesin simbolik MuPAD, memungkinkan akses terhadap kemampuan aljabar komputer. Sebuah paket tambahan, Simulink, menambahkan simulasi grafis multiranah dan desain berdasar-model untuk sistem terlekat dan dinamik. Pada tahun 2004, *MathWorks* mengklaim bahwa MATLAB telah dimanfaatkan oleh lebih dari satu juta pengguna di dunia pendidikan dan industri.

MATLAB diciptakan pada akhir tahun 1970-an oleh Cleve Moler, yang kemudian menjadi Ketua Departemen Ilmu Komputer di Universitas New Mexico. Ia merancang untuk memberikan akses bagi mahasiswa dalam memakai LINPACK dan EISPACK tanpa harus mempelajari Fortran. Karyanya itu segera menyebar ke beberapa universitas lain dan memperoleh sambutan hangat di kalangan komunitas matematika terapan. Jack Little, seorang insinyur, dipertemukan dengan karyanya tersebut selama kunjungan Moler ke Universitas Stanford pada tahun 1983. Menyadari potensi komersialnya, ia bergabung dengan Moler dan Steve Bangert. Mereka menulis ulang MATLAB dalam bahasa pemrograman C, kemudian mendirikan The MathWorks pada tahun 1984 untuk melanjutkan pengembangannya. Pustaka yang ditulis ulang kini dikenal dengan nama JACKPAC. Pada tahun 2000, MATLAB ditulis ulang dengan pemakaian sekumpulan pustaka baru untuk manipulasi matriks, LAPACK.

MATLAB pertama kali diadopsi oleh insinyur rancangan kontrol (yang juga spesialisasi *Little*), tetapi lalu menyebar secara cepat ke berbagai bidang lain. Kini juga digunakan di bidang pendidikan,

khususnya dalam pengajaran aljabar linear dan analisis numerik, serta populer di kalangan ilmuwan yang menekuni bidang pengolahan citra.

2.19. Koefisien Korelasi

Koefisien korelasi adalah nilai yang menunjukkan kuat/tidaknya hubungan linier antar dua variabel. Koefisien korelasi biasa dilambangkan dengan huruf r dimana nilai r dapat bervariasi dari -1 sampai $+1$. Nilai r yang mendekati -1 atau $+1$ menunjukkan hubungan yang kuat antara dua variabel tersebut dan nilai r yang mendekati 0 mengindikasikan lemahnya hubungan antara dua variabel tersebut. Sedangkan tanda $+$ (positif) dan $-$ (negatif) memberikan informasi mengenai arah hubungan antara dua variabel tersebut. Jika bernilai $+$ (positif) maka kedua variabel tersebut memiliki hubungan yang searah. Dalam arti lain peningkatan X akan bersamaan dengan peningkatan Y dan begitu juga sebaliknya. Jika bernilai $-$ (negatif) artinya korelasi antara kedua variabel tersebut bersifat berlawanan. Peningkatan nilai X akan diikuti dengan penurunan Y .

$$r = \frac{cov(x, y)}{\sqrt{D(x) \cdot D(y)^T}}$$

Dimana:

$$cov(x, y) = \frac{1}{s} \sum_{i=1}^s (x_i - \bar{x})(y_i - \bar{y})$$

$$D(x) = \frac{1}{s} \sum_{i=1}^s (x_i - \bar{x})^2$$

2.20. Composite Video

Video komposit (saluran tunggal) adalah transmisi video analog (tanpa audio) yang membawa video definisi standar biasanya pada resolusi 480i atau 576i. Informasi video dikodekan pada satu saluran, tidak seperti S-video berkualitas lebih tinggi (dua saluran) dan video komponen berkualitas lebih tinggi (tiga atau lebih saluran). Video komposit sebagian besar hadir dalam tiga format standar: NTSC, PAL, dan SECAM.

Video komposit dapat dengan mudah diarahkan ke saluran siaran lainnya dengan memodulasi gelombang pembawa RF yang tepat dengannya. Kebanyakan peralatan video analog rumah merekam sinyal dalam format gabungan (kasar): LaserDiscs menyimpan sinyal komposit yang

sebenarnya, sementara format kaset video konsumen (termasuk VHS dan Betamax) dan format kaset komersial dan industri yang lebih rendah (termasuk U-Matic) menggunakan sinyal komposit yang dimodifikasi (umumnya dikenal sebagai warna bawah). Saat diputar, perangkat ini sering memberikan opsi kepada pengguna untuk mengeluarkan sinyal baseband atau memodulasi ke frekuensi VHF atau UHF yang kompatibel dengan TV tuner (yaitu, muncul di saluran TV yang dipilih). Produksi televisi profesional yang tidak dikompresi format video digital kaset video yang dikenal sebagai D-2 (video) merekam secara langsung dan mereproduksi sinyal video komposit NTSC standar, menggunakan pengkodean PCM dari sinyal analog pada pita magnetik.

Dalam aplikasi sehari-hari, sinyal video komposit biasanya dihubungkan menggunakan konektor RCA (phono plug), biasanya berwarna kuning. Hal ini sering disertai dengan konektor merah dan putih (atau hitam) untuk saluran audio kanan dan kiri masing-masing. Konektor BNC dan kabel koaksial berkualitas lebih tinggi sering digunakan di studio televisi profesional dan aplikasi pascaproduksi. Konektor BNC juga digunakan untuk koneksi video komposit pada VCR awal rumah, sering disertai dengan konektor phono atau konektor DIN 5-pin untuk audio. Konektor BNC, pada gilirannya posting tanggal konektor PL-259 yang ditampilkan pada VCR generasi pertama.

2.21. Buck Converter

Tegangan masukan Raspberry Pi 3 adalah 5 volt, sehingga diperlukan penurun tegangan agar tegangan yang diberikan pada Raspberry tidak melebihi atau kurang dari tegangan yang dibutuhkan. *Buck converter* (Gambar 2.8) merupakan konverter penurun tegangan dari *input* ke *output*. *Buck converter* ini menerapkan sistem *Switching Mode Power Supply* (SMPS). Sistem SMPS adalah catu daya elektronik yang menggabungkan pengaturan penggantian (*switching*) untuk mengubah daya listrik secara efisien.

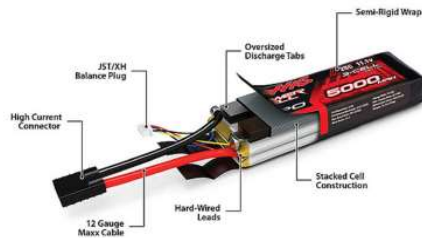
Buck converter yang digunakan adalah IC XL4005. Tegangan masukan pada IC yaitu dari 5 volt sampai 30 volt. Sedangkan tegangan keluaran yang dihasilkan adalah 0.8 volt sampai 30 volt dengan arus maksimal 5 ampere. Tegangan keluaran ini dapat diatur dengan cara memutar *multiturn* yang ada pada *board buck converter*.



Gambar 2.8. Board Buck Converter [13]

2.22. Baterai Lithium Polymer (LiPo)

Baterai *LiPo* seperti pada Gambar.2.9 digunakan sebagai sumber dari rangkaian yang sudah dibuat seperti Arduino dan sensor lainnya. Baterai LiPo tidak menggunakan cairan sebagai elektrolit melainkan menggunakan elektrolit polimer kering yang berbentuk seperti lapisan plastic film tipis. Lapisan ini tersusun berlapis-lapis diantara anoda dan katoda yang mengakibatkan pertukaran ion. Metode ini dapat membuat baterai LiPo dibentuk dengan berbagai macam ukuran. Terdapat kekurangan dari baterai LiPo ini yaitu lemahnya aliran pertukaran ion yang terjadi melalui elektrolit polimer kering. Hal ini menyebabkan penurunan pada *charging* dan *discharging rate*. Dapat diatasi dengan memanaskan baterai sehingga menyebabkan pertukaran ion menjadi lebih cepat, namun metode ini tidak dapat diaplikasikan pada keadaan sehari-hari karena dapat menimbulkan bahaya[16].



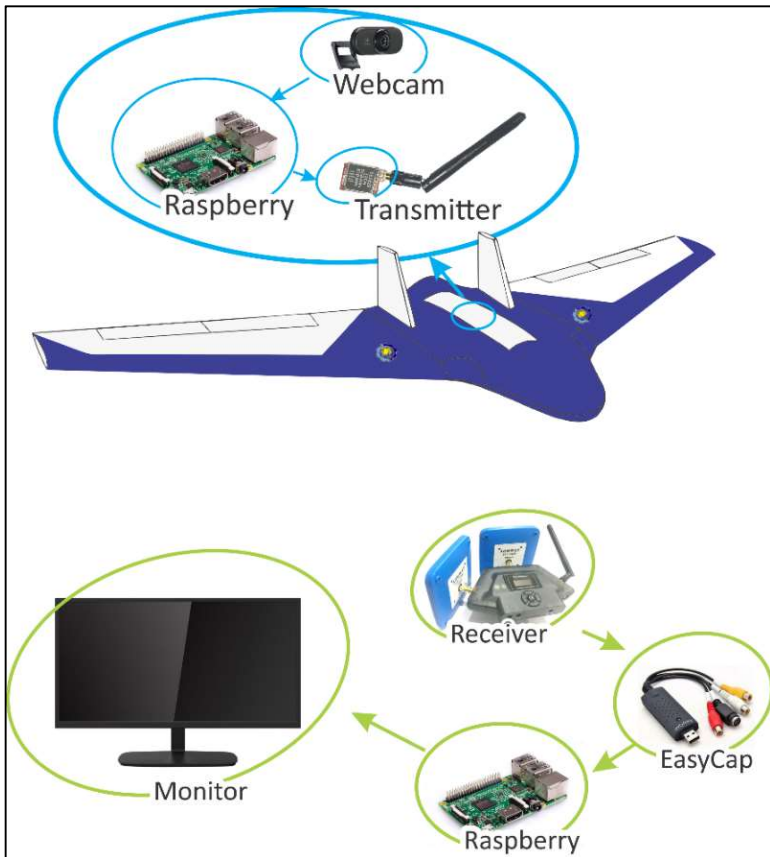
Gambar 2.9. Baterai Lithium Polymer [14]

.....*Halaman ini sengaja dikosongkan*.....

BAB III

PERANCANGAN SISTEM

Pada Bab ini membahas mengenai perancangan sistem pengaman transmisi video pada pesawat tanpa awak sesuai dengan skema alat yang ditunjukkan pada Gambar 3.1 dibawah ini.



Gambar 3.1.Skema Sistem Keseluruhan

Perancangan sistem meliputi spesifikasi sistem yang akan dirancang, identifikasi kebutuhan sistem, perancangan mekanik, perancangan perangkat keras, dan perancangan perangkat lunak dari sistem yang akan dibuat.

3.1. Spesifikasi Sistem

Sebuah sistem pengamanan transmisi video memiliki input sebuah kamera yang dipasang pada pesawat tanpa awak (UAV) yang berfungsi untuk menangkap gambar. Gambar dari kamera kemudian diproses dengan cara diacak sedemikian rupa sehingga tidak dapat dikenali menjadi suatu gambar seperti yang ditangkap kamera untuk selanjutnya ditransmisikan. Pada ground station (penerima) yang mempunyai akses untuk menerima gambar sesuai tangkapan kamera, dilakukan proses rekonstruksi (dekripsi).

Untuk merealisasikan sistem diatas, maka sistem harus memiliki spesifikasi sebagai berikut :

- a. Pemancar diharapkan memiliki dimensi mekanis sekecil mungkin agar dapat diletakkan pada UAV.
- b. Pemancar harus tersuplai oleh sebuah sumber daya yang cukup dan ringan.
- c. Pemancar diharapkan mampu mengirimkan data dengan jarak minimal 100 meter.
- d. Sistem pada perangkat lunak mampu mengenkripsi gambar yang ditangkap kamera dengan kemiripan kurang dari 10 persen. Semakin kecil kemiripan gambar yang terenkripsi dengan gambar asli, semakin baik pula proses pengamanan data transmisi yang dilakukan
- e. Sinyal video yang diterima oleh penerima harus menjadi standar USB yang dapat dikenali oleh *processor*, dalam hal ini Raspberry.
- f. Penerima (*ground station*) mampu merekonstruksi gambar acak yang diterima (dekripsi) menjadi gambar yang menyerupai gambar tangkapan kamera dengan tingkat kemiripan sekurang-kurangnya 90 persen.

3.2. Identifikasi Kebutuhan

Sistem yang akan dibuat harus mampu memenuhi spesifikasi yang ada. Untuk memenuhi spesifikasi tersebut maka terdapat kebutuhan yang harus dipenuhi antara lain :

- a. Perancangan mekanis dilakukan secara minimalis untuk

- mendapatkan dimensi yang sekecil mungkin.
- b. Pemilihan sumber daya pada pemancar diharapkan dengan daya yang cukup dan dimensi serta berat sekecil mungkin.
 - c. Pemancar menggunakan modul *transmitter* TS5828 yang sering digunakan pada UAV.
 - d. Untuk menghasilkan enkripsi dilakukan pengacakan berdasarkan blok gambar, yang urutannya ditentukan oleh Algoritma *LFSR*. Pengacakan ini akan dilakukan oleh mini *PC* Raspberry Pi.
 - e. Pengubahan sinyal video (PAL/NTSC) menjadi standar USB dilakukan oleh *Video Grabber*.
 - f. Penerima (*ground station*) memerlukan algoritma untuk mengubah gambar enkripsi menjadi gambar aslinya.

3.3. Perancangan Mekanik

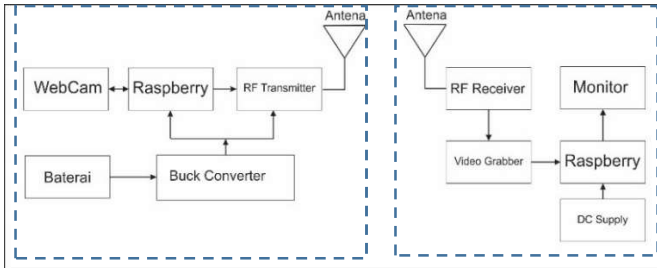
Sistem mekanik yang diharapkan adalah sekecil dan seringan mungkin. Mekanik yang dimaksudkan disini adalah wadah sistem yang didalamnya terdapat raspberry Pi 3 yang terkoneksi dengan *Transmitter* TS5828. Wadah ini dibuat agar *board* Raspberry tidak mendapatkan kontak langsung dari luar sehingga dapat mencegah terjadinya kerusakan elektronik seperti *Electronic Static Discharge* (ESD) maupun terjadi hubungan pendek arus listrik.

Untuk mempersingkat waktu perancangan penulis menggunakan desain yang sudah ada pada situs desain tiga dimensi untuk case raspberry yaitu <https://www.thingiverse.com/>. Desain yang sesuai untuk kemudian diunduh dan dicetak menggunakan 3D printer dan dipasang *board* Raspberry serta TS5828 seperti pada Gambar 3.2



Gambar 3.2. Case Raspberry pada Transmitter

3.4. Perancangan Perangkat Keras



Gambar 3.3. Diagram Blok Perangkat Keras

Perancangan perangkat keras dibutuhkan agar sistem berjalan dengan baik. Perangkat keras yang ada pada sistem ini terdiri dari dua bagian besar. Bagian tersebut adalah *Transmitter* dan *Receiver*. Transmitter akan dipasang pada *UAV* sedangkan Receiver dipasang pada *Ground Station*. Diagram Blok Perangkat Keras seperti terlihat pada Gambar 3.3

3.5. Persiapan Raspberry Pi 3

Agar Raspberry dapat melakukan fungsi yang diinginkan sesuai dengan sistem yang akan dibuat, Raspberry harus sudah terinstall OS Raspbian Jessie, OpenCV, Raspicam, Kate dan GCC

3.5.1. Instalasi Raspbian Jessie

Raspbian Jessie adalah operating sistem dari Raspberry, untuk instalasinya, digunakan sebuah microSD yang telah diformat FAT32 kemudian diinstall atau diflash menggunakan Win32 Disk Imager. Proses ini memakan waktu sekitar 10 menit.

3.5.2. Instalasi OpenCV

Pada terminal ketikkan langkah instalasi OpenCV adalah sebagai berikut :

1. `echo "Installing OpenCV" $version`
2. `mkdir OpenCV`
3. `cd OpenCV`
4. `echo "Removing any pre-installed ffmpeg and x264"`
5. `sudo apt-get -qq remove ffmpeg x264 libx264-dev`
6. `echo "Installing Dependences"`
7. `sudo apt-get -qq install libopencv-dev build-essential
checkinstall cmake pkg-config yasm libjpeg-dev libjasper-dev`

```

libavcodec-dev libavformat-dev libswscale-dev libdc1394-22-
dev libxine2-dev libgstreamer0.10-dev libgstreamer-plugins-
base0.10-dev libv4l-dev python-dev python-numpy libtbb-dev
libqt4-dev libgtk2.0-dev libmp3lame-dev libopencore-amrnb-dev
libopencore-amrwb-dev libtheora-dev libvorbis-dev lib
8. echo "Downloading OpenCV" $version
9. wget -O OpenCV-$version.zip
http://sourceforge.net/projects/opencvlibrary/files/opencv-
unix/$version/opencv-"$version".zip/download
10. echo "Installing OpenCV" $version
11. unzip OpenCV-$version.zip
12. cd opencv-$version
13. mkdir build
14. cd build
15. cmake -D CMAKE_BUILD_TYPE=RELEASE -D
CMAKE_INSTALL_PREFIX=/usr/local -D WITH_TBB=ON -D
BUILD_NEW_PYTHON_SUPPORT=ON -D WITH_V4L=ON -D
INSTALL_C_EXAMPLES=ON -D INSTALL_PYTHON_EXAMPLES=ON -D
BUILD_EXAMPLES=ON -D WITH_QT=ON -D WITH_OPENGL=ON ..
16. make -j2
17. sudo checkinstall
18. sudo ldconfig

```

Proses instalasi membutuhkan waktu 8 jam. Setelah instalasi selesai, maka OpenCV siap digunakan.

3.5.3. Instalasi RaspiCam

Pada terminal, ketikkan *script* dibawah ini untuk menginstal RaspiCam "***sudo raspi-config***". Jika tidak terdapat pilihan "camera", selanjutnya ketikkan "***sudo apt-get update***" dan "***sudo apt-get upgrade***". Masuk ke menu "camera" dan aktifkan kamera dengan cara memilih opsi "*Enable*".

Agar program dapat mengenali kamera, perlu ditambahkan *syntax* "***modprobe bcm2835-v4l2***". Program dapat dijalankan dan dapat mengakses kamera setelah melakukan langkah-langkah diatas.

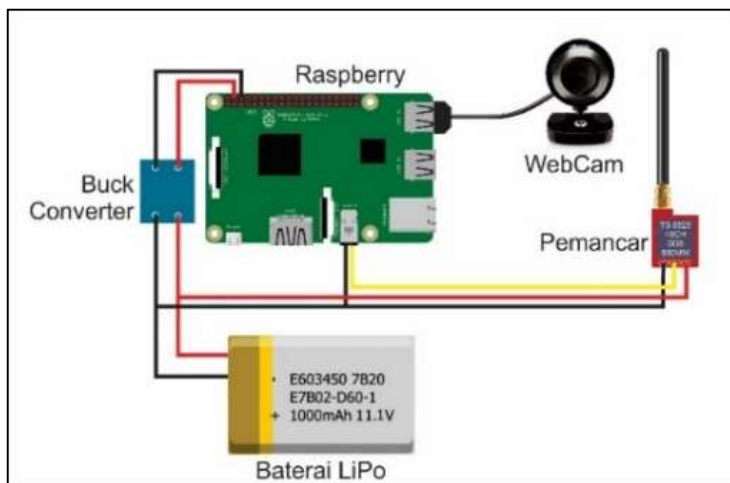
3.5.4. Pengaturan Autostart Program

Autostart diperlukan agar pada waktu raspberry dinyalakan, Raspberry langsung menjalankan program yang telah dibuat. Setelah proses kompilasi program selesai, maka dihasilkan file executable yang dapat dijalankan oleh raspberry. Langkah selanjutnya adalah mengetikkan script "***sudo nano bash.rc***". Mengedit file tersebut dengan mengetikkan "***sudo \home\pi\namaprogram.sh***". Kemudian file disimpan dengan

mengetikkan Ctrl+X. Proses berikutnya adalah mengatur *lxterminal* menjadi program yang pertama kali dibuka ketika Raspberry *boot-up*. Cara mengaktifkannya adalah dengan mengetikkan “*nano ~/.config/lxsession/LXDE-pi/autostart*”. Setelah memasuki file autostart, selanjutnya mengetikkan “*@lxterminal*” pada akhir file. Kemudian save program.

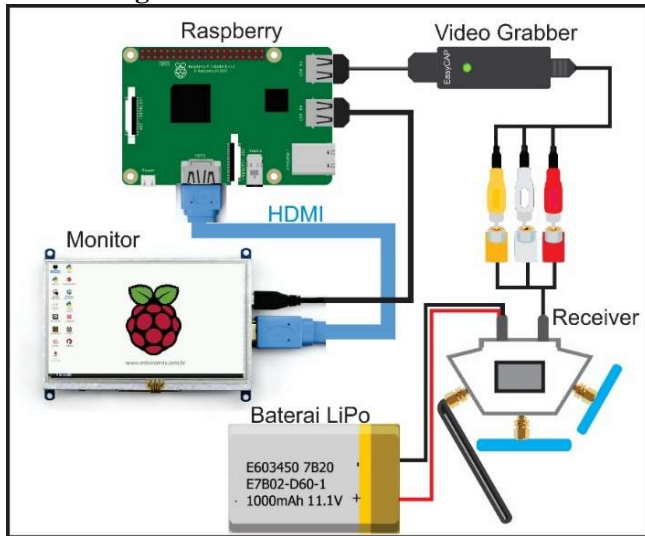
3.6. Perancangan Pemancar

Seperti yang telah dibahas sebelumnya, perangkat keras pemancar akan dipasang pada UAV. Pengkabelan divais pada pemancar ditunjukkan pada Gambar 3.4. Pada *Wiring diagram* Gambar 3.4 terlihat bahwa Webcam terhubung dengan Raspberry menggunakan koneksi USB. Sedangkan Transmitter terhubung pada video/audio output (*Composite*). Sebagai suplai raspberry digunakan kaki GPIO pin 4 dan 6, dimana 4 terhubung dengan output +5V Buck converter dan 6 terhubung dengan *ground*. Suplai pemancar didapat langsung dari baterai, karena spesifikasi tegangan yang dibutuhkan adalah 7-24 Volt.



Gambar 3.4. Diagram Pengkabelan Transmitter

3.6.1 Perancangan Penerima

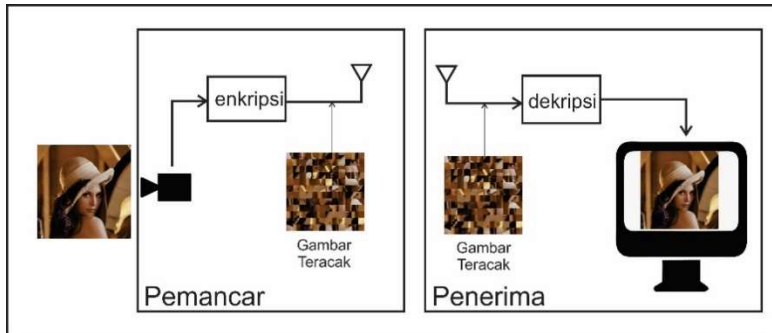


Gambar 3.5. Diagram Pengkabelan Receiver

Penerima terletak pada *ground station*. Pengkabelan seperti ditunjukkan pada Gambar 3.5. Pada sisi Penerima dibutuhkan sebuah RF receiver sebagai penerima sinyal PAL kiriman dari Transmitter di UAV. Sinyal PAL tidak dapat diterima langsung oleh Raspberry. Oleh karena itu diperlukan sebuah video Grabber yang berfungsi untuk mengubah sinyal PAL menjadi standar USB sehingga dapat diolah oleh Raspberry. Data dari Video Grabber diolah oleh perangkat lunak kemudian ditampilkan pada monitor.

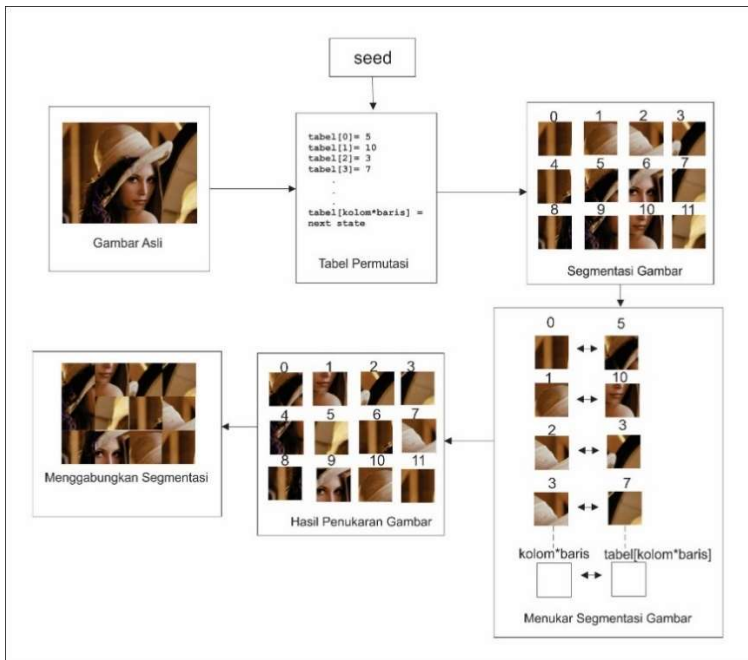
3.7. Perancangan Pengaman Transmisi

Pengamanan transmisi dilakukan oleh perangkat lunak. Perangkat lunak mempunyai fungsi yang krusial pada sistem ini. Perangkat lunaklah yang memproses gambar pada pemancar maupun pada penerima. Pada penerima, perangkat lunak berfungsi untuk mengacak gambar dari kamera sebelum ditransmisikan. Proses ini disebut juga Enkripsi. Sedangkan pada penerima, data transmisi yang berupa gambar acak harus di rekonstruksi sehingga menjadi gambar asli sesuai yang ditangkap oleh kamera. Proses ini disebut Dekripsi. Proses ini dijelaskan pada Gambar 3.6



Gambar 3.6. Diagram Blok Pengamanan Transmisi

3.7.1 Perancangan Enkripsi



Gambar 3.7. Diagram Blok Proses Enkripsi

Enkripsi digunakan untuk mengacak gambar menjadi gambar baru yang sulit dideteksi. Penentuan gambar baru ini berdasarkan pengacakan tempat yang diperoleh dari *Pseudo Random* menggunakan metode *LFSR*. Proses Enkripsi ditunjukkan pada Gambar 3.7. Pertama software menangkap gambar dari kamera. Kemudian dibentuk sebuah tabel permutasi dengan fungsi *Pseudo Random LFSR* dengan input *seed*. Setelah itu, gambar dibagi menjadi beberapa potong sesuai dengan keinginan pengguna. Pada diagram blok dibagi menjadi 4x3 atau 12 potong (nantinya akan dikembangkan hingga 12x10). Kemudian segmentasi gambar akan menghasilkan potongan-potongan gambar baru. Potongan tersebut kemudian ditukar tempatnya berdasarkan table permutasi yang telah dibuat sebelumnya. Setelah pertukaran dilakukan selanjutnya adalah menggabungkan gambar-gambar tadi menjadi gambar enkripsi yang utuh. Gambar inilah yang kemudian di transmisikan.

a. Menerima Gambar

OpenCV – secara spesifik HighGui, menyediakan cara yang mudah untuk mengakses kamera. Pengaksesan kamera hanya perlu menggunakan program seperti dibawah ini :

```
Mat src;
VideoCapture cap;
cap.open(0);
cap >> src;
```

Dengan menggunakan tipe data *VideoCapture*, user sudah dapat menangkap video yang ditangkap oleh kamera. Pembacaan kamera berlangsung pada `cap.open(0)`, artinya cap diisi dengan data kamera yang dibaca oleh processor dengan indeks 0.

b. Menentukan Seed

Seed digunakan selayaknya kunci pada enkripsi. Seed digunakan pada inisialisasi nilai awal *Linear Feedback Shift Register*. Nilai awal ini selanjutnya akan menghasilkan urutan *PseudoRandom*.

Program untuk menentukan *next state* dengan *input seed* adalah sebagai berikut

```
uint16_t LFSR(uint16_t num)
{
    uint16_t bin, buf;
    if (num != 0)
    {
        bin = num;
```

```

        buf = (bin & 1) ^
              ((bin & 4) >> 2) ^
              ((bin & 8) >> 3) ^
              ((bin & 0b100000) >> 5);
        buf = buf << 15;
        bin = bin >> 1;
        bin = bin | buf;
        num = bin;
    }
    return num;
}

```

c. Membuat Tabel Permutasi

Tabel permutasi digunakan untuk menentukan perpindahan segmentasi gambar. Tabel dibuat menggunakan *array*, dimana alamat *array* merupakan koordinat asal gambar, sedangkan isi *array* adalah tujuan dari koordinat gambar

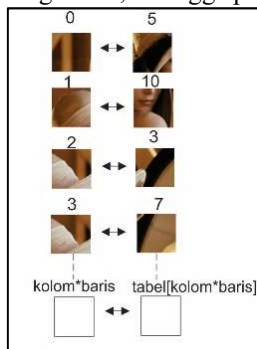
Pembuatan tabel ini memanfaatkan fungsi *LFSR* yang telah dibuat sebelumnya. Yang perlu diperhatikan adalah besar dari tabel itu sendiri, dimana ditentukan oleh banyaknya segmentasi gambar. Atau kolom dikalikan dengan baris. Pada tulisan ini, pembagian segmentasi gambar adalah 16 x 12 yang berarti jumlah segmentasi adalah 192. Program untuk membuat tabel permutasi adalah sebagai berikut.

```

for (int y = 0; y < kolom*baris; y++)
{
    num = LFSR(num);
    tabel[y] = num % (kolom*baris);
}

```

“%” atau modulus digunakan agar hasil perhitungan tabel masih dalam batas jumlah segmentasi gambar, sehingga pertukaran dapat dilakukan



Gambar 3.8. Ilustrasi penukaran gambar

d. Segmentasi Gambar

Segmentasi gambar digunakan untuk mempartisi gambar (dalam hal ini memotong gambar) berdasarkan parameter tertentu, dalam kasus ini adalah berdasarkan pembagian piksel. Pada tulisan ini, segmentasi gambar dilakukan dengan cara membuat *Range of Image*. Pada OpenCV, untuk menentukan ROI dapat dilakukan dengan mengeset koordinat dari gambar.

```
Mat frame;
Mat result[192];
col = 16;
row = 12;
lebar = frame.cols;
panjang = frame.rows;
m = lebar/col
n = panjang/row
for (int j = 0; j < col; j++)
{
    for (int i = 1; i < row; i++)
    {
        tile = (j * row+i);
        Rect ROI((tabel[tile] % row)*n, (tabel[tile] / col)*m, n, m);
        result[j*row+i] = frame(ROI);
    }
}
```

e. Penukaran segmentasi gambar

Setelah segmentasi dilakukan langkah selanjutnya adalah menukar posisi *array* dari *result[]*. Hal ini dapat dilakukan dengan menggunakan perulangan for

```
Mat buffer;
for (int j = 0; j < col*row; j++)
{
    result[j].copyTo(buffer);
    result[tabel[j]].copyTo(result[j]);
    buffer.copyTo(result[tabel[j]]);
}
```

f. Penggabungan gambar

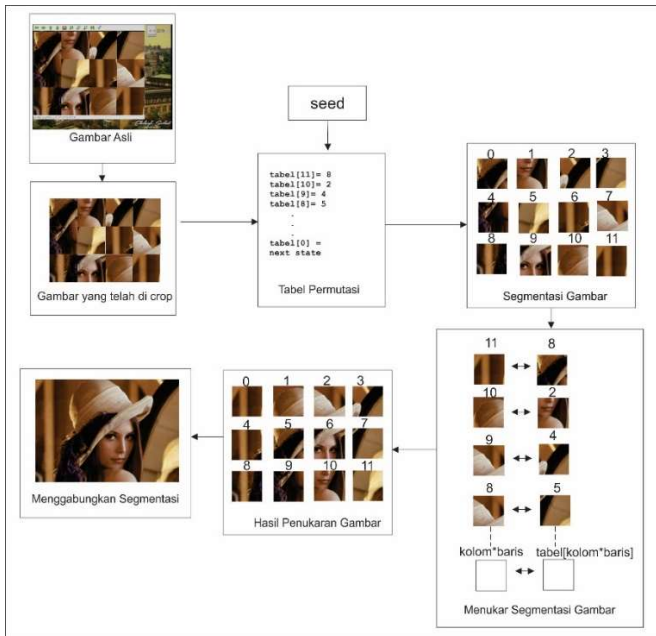
Penggabungan gambar dilakukan agar segmentasi yang telah diacak tadi menjadi satu gambar utuh. Penggabungan dilakukan dengan fungsi *vconcat* dan *hconcat*. Program segmentasi pertukaran dan penggabungan kembali gambar seperti dibawah ini.

```

for (int j = 0; j < col; j++)
{
    tile = (j * row);
    Rect ROI((tabel[tile] % row)*n, (tabel[tile]/row)*m, n, m);
    result[j] = frame(ROI);/-->input
    for (int i = 1; i < row; i++)
    {
        tile = ((j * row) + i);
        Rect ROI((tabel[tile]%row)*n, (tabel2[tile]/row)*m, n, m);
        croppedImage = frame(ROI);/-->input
        hconcat(result[j], croppedImage, result[j]);
    }
    if (j >= 1)
        vconcat(hasil, result[j], hasil);
    else
        result[j].copyTo(hasil);
}

```

3.7.2 Perancangan Dekripsi



Gambar 3.9. Diagram Blok Proses Dekripsi

Prosedur dekripsi melibatkan penerapan algoritma yang digunakan dalam urutan terbalik. Kunci yang sama digunakan untuk proses dekripsi dengan satu-satunya perbedaan adalah bahwa nomor acak yang dihasilkan oleh algoritma digunakan dalam urutan terbalik dibandingkan dengan proses enkripsi. Pada dekripsi juga dilakukan proses *cropping*, karena gambar yang ditransmisikan mempunyai lebar yang lebih kecil dikarenakan adanya *Video Grabber* yang membuat resolusi lebih kecil.

a. **Cropping Gambar**

Proses transmisi data melalui RCA dengan menggunakan EasyCap STK1160 akan mengalami penurunan *pixel* dan *window* yang menampilkan gambar tidak full screen maka dari itu perlu dilakukan *cropping* agar area video yang akan diproses menjadi jelas. Proses *Cropping* ini dilakukan dengan program

```
Rect ROI(0, 28, 512, 384);
crop = source(ROI);
```

b. **Menentukan Seed**

Seed digunakan selayaknya kunci pada dekripsi. *Seed* digunakan pada inisialisasi nilai awal *Linear Feedback Shift Register*. Nilai awal ini selanjutnya akan menghasilkan urutan *PseudoRandom*. Program untuk menentukan *next state* dengan *input seed* adalah sebagai berikut. Agar proses dekripsi berjalan dengan benar. Seed yang digunakan harus sama dengan *seed* yang digunakan pada enkripsi.

```
uint16_t LFSR(uint16_t num)
{
    uint16_t bin, buf;
    if (num != 0)
    {
        bin = num;
        buf = (bin & 1) ^ ((bin & 4) >> 2) ^ ((bin & 8) >> 3) ^ ((bin & 0b100000) >> 5);
        buf = buf << 15;
        bin = bin >> 1;
        bin = bin | buf;
        num = bin;
    }
    return num;
}
```

c. **Membuat Tabel Permutasi**

Setelah dibuat fungsi untuk mencari *next state Pseudorandom*, selanjutnya digunakan array untuk membentuk sebuah tabel penukaran.

Hal ini dilakukan untuk mempermudah dan mempercepat proses komputasi. Jadi tabel yang dalam program komputer diwakili oleh sebuah array akan diisi dengan nilai *next state*. Yang mana merupakan destinasi penukaran data. Penukaran data dilakukan hingga semua data (*width*) terisi tabel sesuai dengan urutannya. Pada kasus ini, *width* atau lebar data adalah 16 x 12, karena gambar nantinya akan dibagi menjadi 16 kekanan dan 12 kebawah.

```
for (int y = kolom*baris-1 ; y >= 0; y--)
{
    num = LFSR(num);
    tabel[y] = num % (kolom*baris);
}
```

d. Segmentasi Gambar

Segmentasi gambar digunakan untuk mempartisi gambar (dalam hal ini memotong gambar) berdasarkan parameter tertentu, dalam kasus ini adalah berdasarkan pembagian piksel. Pada tulisan ini, segmentasi gambar dilakukan dengan cara membuat *Range of Image*. Pada OpenCV, untuk menentukan ROI dapat dilakukan dengan mengatur koordinat dari gambar.

```
Mat frame;
Mat result[192];
col = 16;
row = 12;
lebar = frame.cols;
panjang = frame.rows;
m = lebar/col
n = panjang/row
for (int j = 0; j < col; j++)
{
    for (int i = 1; i < row; i++)
    {
        tile = (j * row+i);
        Rect ROI((tabel[tile] % row)*n, (tabel[tile] / col)*m, n, m);
        result[j*row+i] = frame(ROI);
    }
}
```

e. Penukaran segmentasi gambar

Setelah segmentasi dilakukan langkah selanjutnya adalah menukar posisi *array* dari *result[]*. Hal ini dapat dilakukan dengan menggunakan perulangan for

```

Mat buffer;
for (int j = 0; j < col*row; j++)
{
    result[j].copyTo(buffer);
    result[tabel[j]].copyTo(result[j]);
    buffer.copyTo(result[tabel[j]]);
}

```

f. Penggabungan gambar

Penggabungan gambar dilakukan agar segmentasi yang telah diacak menjadi satu gambar utuh. Penggabungan dilakukan dengan fungsi `vconcat` dan `hconcat`. Program segmentasi pertukaran dan penggabungan kembali gambar seperti dibawah ini.

```

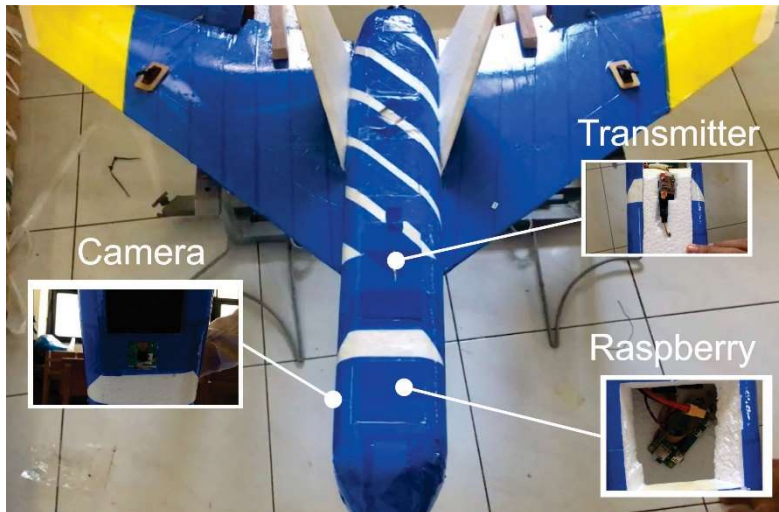
for (int j = 0; j < col; j++)
{
    tile = (j * row);
    Rect ROI((tabel[tile] % row)*n, (tabel[tile]/row)*m, n, m);
    result[j] = frame(ROI);/-->input
    for (int i = 1; i < row; i++)
    {
        tile = ((j * row) + i);
        Rect ROI((tabel[tile]%row)*n, (tabel2[tile]/row)*m, n, m);
        croppedImage = frame(ROI);/-->input
        hconcat(result[j], croppedImage, result[j]);
    }
    if (j >= 1)
        vconcat(hasil, result[j], hasil);
    else
        result[j].copyTo(hasil);
}

```

3.8. Penempatan pada *Unmanned Aerial Vehicle (UAV)*

Sistem akan diletakkan pada pesawat tanpa awak (UAV). Pesawat ini kemudian diterbangkan dan dikontrol secara *mobile* menggunakan *remote control*. Berikut adalah penempatan komponen pada pesawat

Kamera diletakkan dibagian bawah pesawat, menghadap kebawah yang dimaksudkan agar dapat menangkap gambar dibawah saat UAV terbang. Transmitter diletakkan diatas, agar lebih mudah dalam proses transmisi.



Gambar 3.10. Pemasangan sistem pada UAV

3.9. Langkah Penggunaan Alat

Langkah penggunaan alat dibutuhkan agar pengguna dapat mengoperasikan alat dengan benar dan menghindari malfungsi alat. Langkah penggunaan alat adalah sebagai berikut :

a. Prosedur Keselamatan

Sebelum menggunakan alat, pengguna perlu mengetahui prosedur keamanan yang benar. Keamanan yang harus diperhatikan meliputi ujung-ujung yang tajam (*sharp edge*), Baterai Lithium, *Electrical Shock*, dan *Short Circuit*.

b. Persiapan Alat

Sebelum diterbangkan, alat harus sudah terpasang pada pesawat. Memastikan bahwa baterai yang akan dipakai dalam kondisi *full charged* dan titik setimbang pesawat telah tercapai.

Langkah selanjutnya adalah melakukan pengetesan terhadap sistem, mengeset pesawat dalam kondisi *on*, memastikan pemancar *on*, dan menyalakan penerima sesuai konfigurasi yang telah ditentukan. Menjalankan program dekripsi dan memastikan proses pengiriman video dan enkripsi berjalan dengan benar.

c. Pengamatan Hasil

Setelah langkah persiapan dilakukan, selanjutnya adalah proses menerbangkan pesawat, dimana pesawat akan diluncurkan dengan tangan kemudian dikontrol dengan *remote*. Ketika pesawat sudah berada pada posisi diatas barulah data dapat diamati. Data yang berupa hasil dekripsi terlihat pada monitor di sistem penerima. Untuk melihat hasil enkripsi, dilakukan dengan menggunakan FPV yang tidak dilengkapi dengan dekripsi.

.

.....*Halaman ini sengaja dikosongkan*.....

BAB IV PENGUJIAN DAN ANALISIS

Berdasarkan spesifikasi dan perancangan sistem yang telah dijelaskan pada bab sebelumnya, selanjutnya dilakukan pengujian dan analisis terhadap perilaku sistem. Tujuan dari pengujian adalah untuk mengetahui performansi sistem hasil perancangan. Jika performa belum sesuai, maka perlu dianalisis untuk pengembangan lebih lanjut.

4.1. Pengujian *Hardware*

Pengujian *Hardware* bertujuan agar sistem berjalan sesuai yang diinginkan sebelum deprogram. Pengujian ini meliputi pengujian jarak transmisi

4.1.1. Pengujian Daya Sistem Transmisi

Pengujian ini bertujuan untuk mengetahui daya yang dibutuhkan oleh Transmisi. Pengujian dilakukan dengan mengukur arus yang mengalir pada rangkaian. Daya dapat diperoleh dengan Persamaan 4.1

$$P = V \times I$$

Dimana :

P = Daya yang dibutuhkan (watt)

V = Tegangan (Volt)

I = Arus (ampere)

Pengukuran dilakukan dengan cara menghubungkan sistem dengan baterai yang di seri dengan ampere meter. Hasil pengukuran seperti terlihat pada Tabel 4.1 berikut

Tabel 4.1 Hasil Pengukuran Daya Sistem

No	Objek ukur	Hasil Pengukuran		Daya (Watt)
		Tegangan (Volt)	Arus (Ampere)	
1	Raspberry + Raspycam	5	259	1.295
2	Raspberry + Raspycam + Buck Converter + Transmitter	12	758	9.096

4.1.2. Pengujian Jarak Transmisi

Pengujian ini bertujuan untuk mengetahui jarak pengiriman data yang baik. Untuk melakukan pengujian ini dibutuhkan Raspberry yang telah dihubungkan dengan transmitter, dan sebuah receiver yang mempunyai *display* berupa monitor. Dengan mengatur jarak sesuai dengan tabel kemudian diamati hasil penerimaan data dan dimasukkan hasilnya seperti yang ditunjukkan pada Tabel 4.2 berikut

Tabel 4.2. Pengujian Transmisi Video

No	Jarak (m)	Keterangan	Hasil
1	100	Teknik Industri – Danau 8	Baik
2	200	Teknik Industri – Taman Refleksi	Baik
3	300	Teknik Industri – Minimarket K1	Baik
4	400	Teknik Industri – Bundaran Biologi	Baik
5	500	Jl. Tek. Geodesi – Bundaran Biologi	Baik
6	600	Tikungan T. Mesin – Bundaran Biologi	Noise

- **Analisis Hasil Pengujian**

Berdasarkan Tabel 4.1 dapat dilihat bahwa data mengalami noise pada 600m, namun noise tersebut berkurang saat arah dari antenna sesuai dengan sumber transmisi. Ini dikarenakan antenna yang dipakai adalah tipe batang, yang jangkauan sinyalnya kurang besar. Hal lain yang mempengaruhi adalah adanya obstacle/halangan.

Penulis sebelumnya telah mencoba mentransmisikan dengan jarak 100 meter, dari lantai 2 laboratorium A206 ke jalan teknik industri. Namun sinyal yang diperoleh juga hanya hitam putih dan memiliki banyak sekali *noise*

4.2. Pengujian Software (Enkripsi)

4.2.1. Pengujian Enkripsi dengan Gambar Statis pada Komputer

Pengujian ini bertujuan untuk menguji keberhasilan enkripsi dan pengaruh *seed* (benih) terhadap hasil enkripsi. Pengujian dilakukan pada file gambar diam terlebih dahulu agar lebih memudahkan dalam pemrograman berikutnya. Untuk melakukan pengujian ini digunakan *Software* Visual Studio. Berikut hasil pengujian enkripsi yang dilakukan dengan memberikan seed yang berbeda-beda, seperti yang ditunjukkan pada Gambar 4.1, hingga Gambar 4.12.

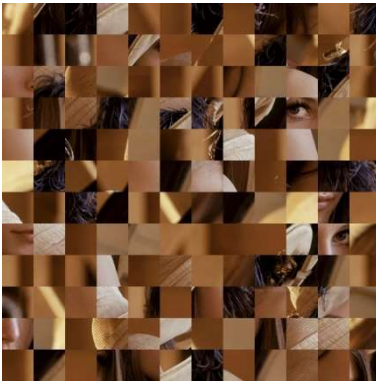
- Kasus 1



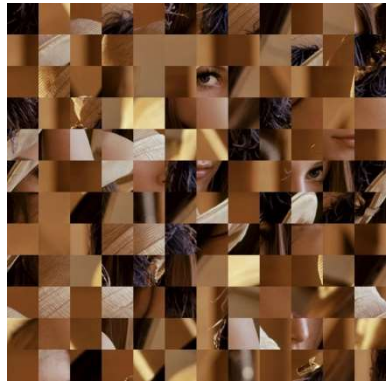
Gambar 4.1. Gambar Asli



Gambar 4.2. Enkripsi *seed* 44257



Gambar 4.3. Enkripsi *seed* 1234



Gambar 4.4. Enkripsi *seed* 1111

- **Kasus 2**



Gambar 4.5. Gambar Asli



Gambar 4.6. Enkripsi seed 44257



Gambar 4.7. Enkripsi seed 1234



Gambar 4.8. Enkripsi seed 1111

- **Kasus 3**



Gambar 4.9. Gambar Asli



Gambar 4.10. Enkripsi seed 44257



Gambar 4.11. Enkripsi seed 1234



Gambar 4.12. Enkripsi seed 1111

- **Analisis Hasil Pengujian**

Berdasarkan data diatas diketahui bahwa berbagai *seed* (benih) yang diberikan pada *LFSR* akan mempengaruhi pengacakan. Terbukti bahwa masing-masing hasil enkripsi mempunyai *output* yang berbeda-beda sesuai dengan perbedaan *seed* yang digunakan

Hasil perhitungan similaritas dari masing-masing enkripsi terhadap gambar aslinya ditunjukkan pada Tabel 4.2, Tabel 4.3 dan Tabel 4.4

Tabel 4.3. Similaritas Gambar pada Kasus 1

Seed	Nilai Koeffisien Korelasi	Nilai Similaritas
44257	0.05261677	5.261676984
1234	0.007384094	0.738409394
1111	-0.039455548	3.94555479
Rata-rata		3.315213722

Tabel 4.4. Similaritas Gambar pada Kasus 2

Seed	Nilai Koeffisien Korelasi	Nilai Similaritas
44257	0.042408179	4.240817872
1234	-0.105766526	10.5766526
1111	-0.04489818	4.489818024
Rata-rata		6.435762833

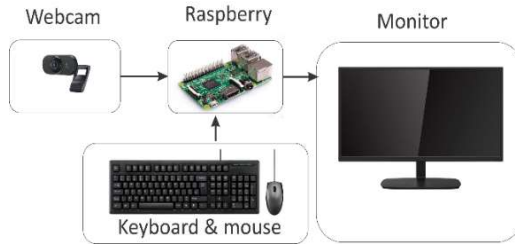
Tabel 4.5. Similaritas Gambar pada Kasus 3

Seed	Nilai Koeffisien Korelasi	Nilai Similaritas
44257	-0.014778128	1.477812837
1234	-0.006259771	0.625977084
1111	-0.040760961	4.076096058
Rata-rata		2.059961993

Dari ketiga kasus diatas maka didapatkan bahwa similaritas rata-rata hasil enkripsi adalah 3.93698 %

4.2.2. Pengujian Enkripsi dengan Kamera pada Raspberry

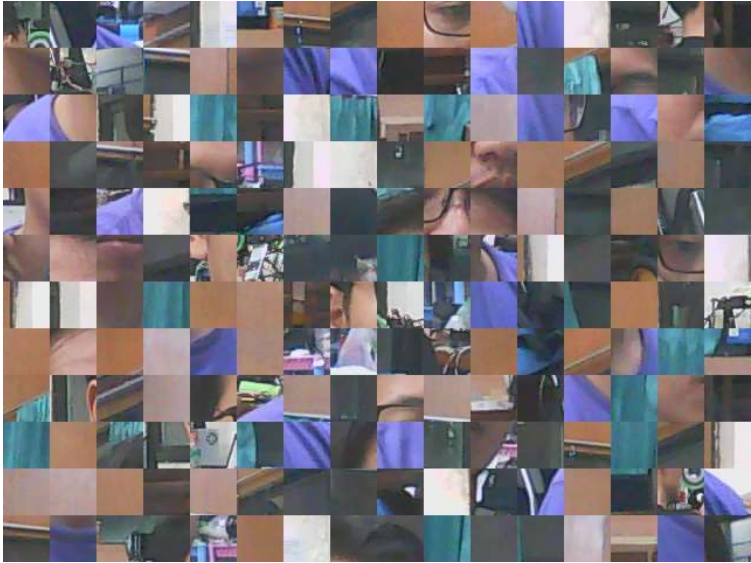
Pengujian ini bertujuan untuk mensimulasikan program ketika mendapatkan input dengan kamera dan mengenkripsikannya pada pengujian ini Raspberry dirangkai sesuai Gambar 4.13. Kemudian program dijalankan



Gambar 4.13. Diagram Blok Enkripsi



Gambar 4.14. Gambar yang tertangkap kamera



Gambar 4.15. Hasil Enkripsi Gambar dari kamera

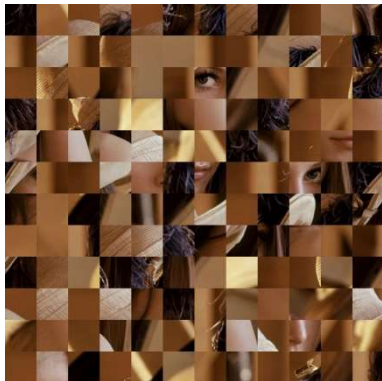
- **Analisis Hasil Pengujian**

Seperti Enkripsi pada Gambar, pada kamera juga akan tergantung pada seed yang digunakan. Pada kamera digunakan pembagian 12 x16 blok. Hasil Enkripsi menggunakan seed 44257.

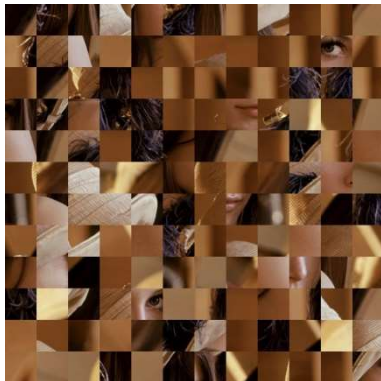
4.3. Pengujian Software Dekripsi dengan File Gambar

Pengujian ini bertujuan untuk mengembalikan gambar yang telah terenkripsi menjadi bentuk aslinya. Dan untuk membuktikan bahwa, seed yang berbeda antara enkripsi dan dekripsi tidak akan menghasilkan dekripsi yang benar.

- Kasus 1



Gambar 4.16. Gambar Enkripsi



Gambar 4.17. Dekripsi seed 44257



Gambar 4.18. Dekripsi seed 1234



Gambar 4.19. Dekripsi seed 1111

-Kasus 2



Gambar 4.20. Gambar Enkripsi



Gambar 4.21. Dekripsi seed 44257



Gambar 4.22. Dekripsi seed 1234



Gambar 4.23. Dekripsi seed 1111

- **Analisis Hasil Pengujian**

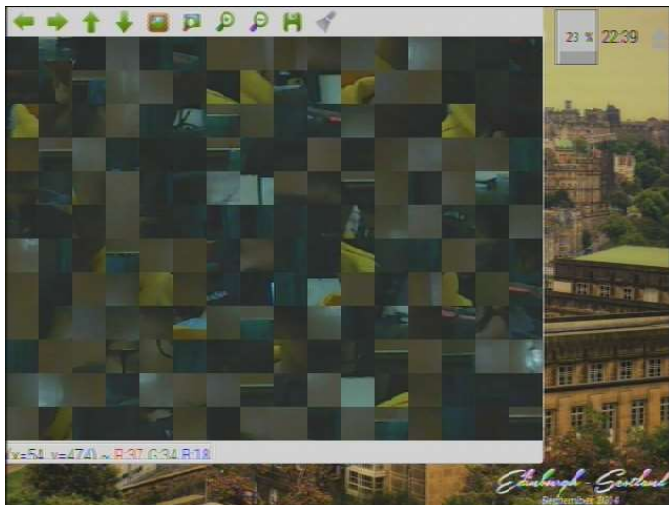
Seperti pada Enkripsi pada dekripsi juga diperlukan *seed*, *seed* digunakan untuk membuat urutan pseudorandom. Jika *seed* yang digunakan pada dekripsi tidak sama dengan yang digunakan pada enkripsi, maka gambar tidak akan kembali seperti semula. Oleh karena itu sistem ini dikatakan aman. Karena pada sisi dekripsi harus mengetahui metode pembentukan *Pseudorandom* dan juga *seed* yang digunakan.

4.4. Pengujian Software Dekripsi dengan Kamera

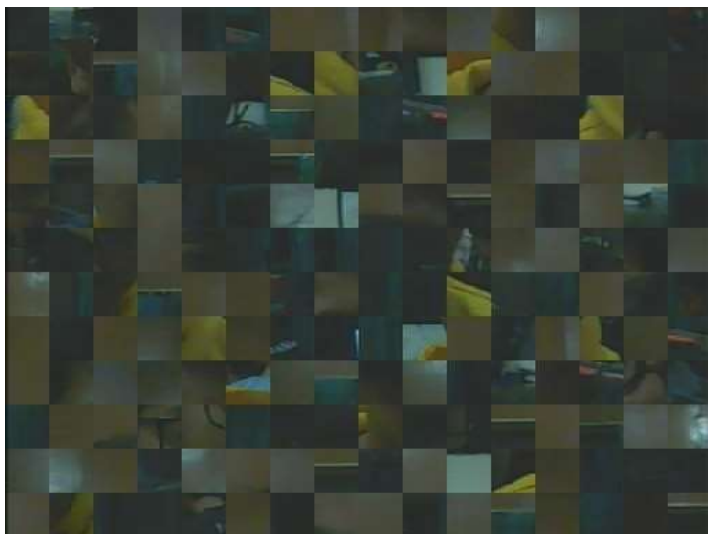
Pengujian ini bertujuan untuk menguji sistem dekripsi secara keseluruhan. Sinyal dari Transmitter yang diterima oleh *receiver* diubah menjadi standar USB oleh Easycap. Data gambar yang telah dienkripsikan, perlu diuji apakah dapat didekripsikan dengan benar.



Gambar 4.24. Gambar tangkapan WebCam



Gambar 4.25. Gambar yang diterima *Receiver*



Gambar 4.26. Gambar hasil Crop



Gambar 4.27. Gambar hasil Dekripsi

- **Analisis Hasil Pengujian**

Dekripsi yang dilakukan kali ini lebih rumit dibanding sebelumnya, dikarenakan kita harus melakukan proses *cropping* terlebih dahulu agar gambar yang akan di dekripsi hanya gambar yang diinginkan. Namun pada proses ini masih terkendala tampilan hasil dekripsi masih memiliki *noise*.

Kemudian ketiga gambar dinilai tingkat kemiripannya dengan koefisien korelasi menggunakan Matlab dan didapatkan hasil pada Tabel 4.4 dibawah ini.

Tabel 4.4. Similaritas Gambar Enkripsi dan Dekripsi terhadap Gambar Asli

Objek Uji	Nilai Koeffisien Korelasi	Nilai Similaritas (%)
Gambar Asli – Gambar Enkripsi	-0.034068912	3.406891212
Gambar Asli – Gambar Dekripsi	0.982434128	98.2434128

4.5. Pengujian Kemiripan Gambar Hasil Dekripsi dengan Gambar Asli

Pengujian ini bertujuan untuk menguji kualitas hasil dekripsi. Pengujian dilakukan dengan menggunakan Matlab. Dua gambar berupa gambar asli yang diperoleh langsung dari kamera dan Gambar transmisi yang di dekripsikan dibandingkan menggunakan koefisien korelasi

```
>> A=imread('D:\1.TA\Gambar\originalresize.jpg');
>> B=imread('D:\1.TA\Gambar\DEKRIPSI.jpg');
>> a=rgb2gray(A);
>> b=rgb2gray(B);
>> corr2(b,a)

ans =

    783/797

fx >>
```

Gambar 4.28. Screenshoot MatLab

Hasil Koefisien Korelasi adalah 0,9824 atau dapat dikatakan kemiripan gambar adalah 98,24%

4.6. Pengujian pada UAV

Pengujian ini dimaksudkan untuk menguji kelayakan sistem ketika diaplikasikan langsung pada UAV. Komponen transmisi mula-mula diletakkan pada UAV, dan dilakukan pengetesan sebelum pesawat diterbangkan, setelah sistem berjalan sesuai yang diharapkan, barulah pesawat diterbangkan. Hasil pengujian tersebut berupa gambar yang ditangkap oleh kamera yang terpasang pada pesawat. Lokasi pengambilan data adalah di Gedung Robotika ITS



Gambar 4.29 Sinyal gambar yang diterima



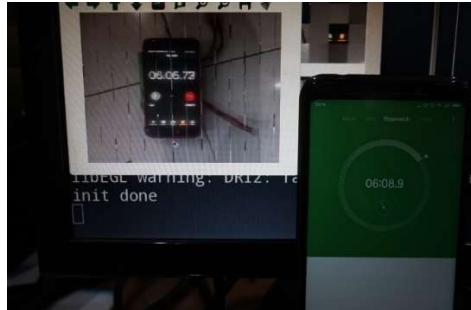
Gambar 4.30. Sinyal gambar yang terenkripsi

4.7. Pengujian Waktu Tunda (Latensi)

Pengujian ini digunakan untuk mengetahui waktu tunda atau latensi yang diperlukan untuk mentransmisi dan mendekripsi video. Pengujian dilakukan dengan cara meletakkan dua *stopwatch* yang diset sama pada kamera di pesawat dan yang lainnya pada ground station di sisi penerima. Selisih antara kedua *stopwatch* selanjutnya adalah latensi atau waktu tunda sistem



Gambar 4.31. Pengesetan stopwatch



Gambar 4.32. Selisih stopwatch

Selisih antara pengiriman dan penerimaan adalah 06.08.9detik - 06.06.7detik = 2,2 detik

.....*Halaman ini sengaja dikosongkan*.....

BAB V

PENUTUP

Hasil dari perancangan dan penelitian tugas akhir dirangkum dan dirumuskan kesimpulan. Kesimpulan ini menerangkan hasil dari pengujian yang telah dilakukan,

Selama proses perancangan dan penelitian, terdapat banyak kendala yang dihadapi. Kendala tersebut telah penulis rangkum dan dirumuskan dalam bentuk saran untuk penyempurnaan dan penelitian lebih lanjut.

5.1. Kesimpulan

Berdasarkan percobaan yang telah dilakukan pada pelaksanaan tugas akhir ini didapat beberapa kesimpulan sebagai berikut:

1. *Seed* yang diberikan pada *Linear Feedback Shift Register* akan mempengaruhi pola pengacakan enkripsi dengan similaritas rata-rata adalah 3,94 %
2. *Seed* yang diberikan pada enkripsi dan dekripsi harus sama agar gambar dapat di rekonstruksi dengan benar.
3. Kemiripan gambar asli dengan rekontruksi gambar pada dekripsi memiliki kemiripan 98,24 %. Sedangkan kemiripan gambar asli dengan gambar yang terenripsi adalah sebesar 3,41%.
4. Latensi pengiriman data adalah sebesar 2,2 detik

5.2. Saran

Sebagai sarana pengembangan *Video Transmision Security* ini, maka terdapat beberapa saran dari penulis berdasarkan hasil yang diperoleh saat percobaan, yaitu sebagai berikut:

1. Untuk mendapatkan similaritas enkripsi secara maksimal, pembagian segmentasi data dapat diperbanyak, pada tulisan ini pembagian hanya dilakukan 16x12. Jika diperbanyak lagi, similaritas akan semakin kecil, dengan kata lain similaritas enkripsi lebih baik.
2. Perlu ditambahkan sebuah pemroses data langsung dari kamera menuju ke transmitter tanpa menggunakan raspberry agar latensi lebih kecil dan pemrosesan data lebih cepat.
3. Transmitter dapat menggunakan tipe yang mempunyai frekuensi lebih besar, agar jarak transmisi lebih jauh.

.....*Halaman ini sengaja dikosongkan*.....

DAFTAR PUSTAKA

- [1] P. A. Lloyd, "The Use of Drones During The Vietnam War.," 01 May 2018. [Online]. Available: <http://peteralanlloyd.com/the-use-of-drones-during-the-vietnam-war/>. [Accessed 17 Juli 2018].
- [2] F. Baylin, R. Maddox and J. M. Cormac, World satellite TV and scrambling methods: the technicians handbook, Baylin Publications , 1993.
- [3] The RAND Corporation, "A Million Random Digits," 30 March 2017. [Online]. Available: https://www.rand.org/pubs/monograph_reports/MR1418/index2.html. [Accessed 26 May 2018].
- [4] "computing, Numerical recipes: the art of scientific," Cambridge University Press, 2016.
- [5] P. Geremia, "Cyclic Redundancy Check Computation: An Implementation Using the TMS320C54x," Texas Instrument, Texas, 1999.
- [6] DronZon, "10 Best FPV Drone Goggles For A Terrific Flying Experience," 01 Mei 2018. [Online]. Available: <http://www.dronezon.com/drone-reviews/fpv-goggles-for-drones-to-experience-the-thrill-of-flying/>. [Accessed 2018 Juli 18].
- [7] Foxtech, "TS5823 5.8G 200mw 32CH Super Mini VTX," Foxtech, [Online]. Available: <https://www.foxtechfpv.com/ts5823-58g-200mw-32ch-super-mini-vtx-p-1454.html>. [Accessed 20 Juni 2018].
- [8] A. Schalk, "Tv viewing and recording solutions for Linux," 13 04 2016. [Online]. Available: <http://easycap.blogspot.co.id/>. [Accessed 2 03 2018].
- [9] SourceForge, "EasyCAP DC60 Driver," SourceForge, [Online]. Available: <https://sourceforge.net/projects/easycapdc60/>. [Accessed 8 Juni 2018].
- [10] Cinescopophilia, "That's Not a Drone.... This Is a Drone: The Stationair VTOL UAV Professional Drone;," Cinescopophilia, [Online]. Available: <https://cinescopophilia.com/thats-not-a-drone-this-is-a-drone-the-stationair-vtol-uav-professional-drone/>. [Accessed 20 Juni 2018].

- [11] Raspberry, "Raspberry," [Online]. Available: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. [Accessed 20 Juni 2018].
- [12] B. Eckel, Thinking in C++, Vol. 1: Introduction to Standard C++, 2nd Edition, Prentice Hall, 2000.
- [13] G. Bradsky and A. Kaehler, Learning OpenCV, OReilly Media, 2015.
- [14] Gearbest, "LM2596 DC to DC Step - Down Transformer Module Power Buck Converter Voltage Regulator Board," GearBest, [Online]. Available: https://www.gearbest.com/development-boards/pp_51010.html. [Accessed 20 Juni 2018].
- [15] D. a. Fun, "Batteries For UAV," [Online]. Available: <http://www.dronesarefun.com/BatteriesForUAV.htm>. [Accessed 20 Juni 2018].
- [16] "C++," Wikipedia, [Online]. Available: https://id.wikipedia.org/wiki/C%2B%2B#cite_ref-2. [Accessed 25 May 2018].

LAMPIRAN A

Program Enkripsi

```
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include <iostream>

#include <stdio.h>
#include <wiringPi.h>

using namespace std;
using namespace cv;

uint16_t LFSR(uint16_t num)
{
    uint16_t bin, fery;

    if (num != 0)
    {
        bin = num;
        fery = (bin & 1) ^ ((bin & 4) >> 2) ^ ((bin &
8) >> 3) ^ ((bin & 0b100000) >> 5);
        fery = fery << 15;
        bin = bin >> 1;
        bin = bin | fery;
        num = bin;
    }
    return num;
}

int main(int argc, char** argv)
{
    Mat src, dst, garis, ball, frame;
    int pembagian;
    float k = 0, l = 0;
```

```

int tabel[200], tabel2[200], num;
num = 44257;
int lebar = 192;
for (int y = 0; y < lebar; y++)
{
    tabel2[y] = y;
}

for (int y = 0; y < lebar; y++)
{
    num = LFSR(num);
    tabel[y] = num % (lebar);
}

//ENKRIPSI-----
int buffer;
for (int y = 0; y < lebar; y++)
{
    buffer = tabel2[tabel[y]];
    tabel2[tabel[y]] = tabel2[y];
    tabel2[y] = buffer;
}

```

```

VideoCapture cap;
cap.open(0);
cap >> frame;
pembagian = 40;
int tile;
Mat result[50];
Mat hasil;
Mat croppedImage;
int lebar1, panjang;
lebar1 = frame.cols;
panjang = frame.rows;
float n, m;

int bagi_lebar, bagi_panjang;

```



```

bagi_lebar = lebar1/pembagian;
bagi_panjang =panjang/pembagian;
n = pembagian;
m = pembagian;
namedWindow("Enkripsi", CV_WINDOW_NORMAL);
cvResizewindow("Enkripsi", 576,432);
moveWindow("Enkripsi",0,0);
while (1)
{
    cap >> frame;
    //TAMPILAN ENKRIPSI ----output hasil
    for (int j = 0; j < bagi_panjang; j++)
    {
        tile = (j * bagi_lebar);
        Rect ROI((tabel2[tile] % bagi_lebar
)*n, (tabel2[tile] / bagi_lebar)*m, n, m);
        result[j] = frame(ROI);//-->input
        for (int i = 1; i < bagi_lebar; i++)
        {
            tile = ((j * bagi_lebar) + i);
            Rect ROI((tabel2[tile] %
bagi_lebar)*n, (tabel2[tile] / bagi_lebar)*m, n, m);
            croppedImage = frame(ROI);/--
>input
            hconcat(result[j], croppedImage,
result[j]);
        }

        if (j >= 1)
            vconcat(hasil, result[j], hasil);

        else
            result[j].copyTo(hasil);
    }

    imshow("Enkripsi", hasil);

```

```
cvResizeWindow("Enkripsi", 576,432);

if (cv::waitKey(5) == 's')
{
    imwrite("Enkripsi.jpg", hasil);
    imwrite("original.jpg", frame);

    return-1;
}

//waitKey(0);
}
return 0;
}
```

LAMPIRAN B

Program Dekripsi

```
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include <iostream>
#include <stdio.h>

using namespace std;
using namespace cv;

uint16_t LFSR(uint16_t num)
{
    uint16_t bin, fery;

    if (num != 0)
    {
        bin = num;
        fery = (bin & 1) ^ ((bin & 4) >> 2) ^ ((bin &
8) >> 3) ^ ((bin & 0b100000) >> 5);
        fery = fery << 15;
        bin = bin >> 1;
        bin = bin | fery;
        num = bin;
    }
    return num;
}

int main(int argc, char** argv)
{
    Mat src, crop, garis, ball, frame;
    float k = 0, l = 0;
    int A,B,C,D;
    VideoCapture cap;
    cap.open(0);
```

```

A=0;
B=32;
C=512;
D=413;
    while (1)
    {

cap >> src;
Rect ROI(0, 29, 512, 384);
crop = src(ROI);

int na;
int ma;

int tabel[200], tabel2[200], num,tabel3[2000];
num = 44257;
int lebar = 192;

int tile;
Mat result[15];
Mat hasil,hasil2;
Mat croppedImage;

num = 44257;
for (int y = 0; y < lebar; y++)
{
    num = LFSR(num);
    tabel[y] = num % (lebar);
}

for (int y = 0; y < lebar; y++)
{
    tabel3[y] = y;
}

for (int y = lebar - 1; y >= 0; y--)
{

```

```

        int buffer = tabel3[tabel[y]];
        tabel3[tabel[y]] = tabel3[y];
        tabel3[y] = buffer;
    }

    int lbr = crop.cols;
    int pjg = crop.rows;

    na = lbr / 16;
    ma = pjg / 12;

    for (int j = 0; j < 12; j++)
    {
        tile = (j * 16);
        Rect ROI((tabel3[tile] % 16)*na,
(tabel3[tile] / 16)*ma, na, ma);
        result[j] = crop(ROI);//input

        for (int i = 1; i < 16; i++)
        {
            tile = ((j * 16) + i);
            Rect ROI((tabel3[tile] % 16)*na,
(tabel3[tile] / 16)*ma, na, ma);
            croppedImage = crop(ROI);//input
            hconcat(result[j], croppedImage,
result[j]);
        }

        if (j >= 1)
            vconcat(hasil2, result[j],
hasil2);//output
        else
            result[j].copyTo(hasil2);//output
    }

```

```

namedWindow("ASLI2", CV_WINDOW_AUTOSIZE);
moveWindow("ASLI2", 20,0);
imshow("ASLI2", src);
namedWindow("CROP", CV_WINDOW_AUTOSIZE);
imshow("CROP", crop);
namedWindow("DEKRIPSI", CV_WINDOW_AUTOSIZE);
imshow("DEKRIPSI", hasil2);

    if (cv::waitKey(5) == 's')
    {
        imwrite("DEKRIPSI.jpg", hasil2);
        imwrite("asli.jpg", src);
        imwrite("crop.jpg", crop);
return-1;
    }

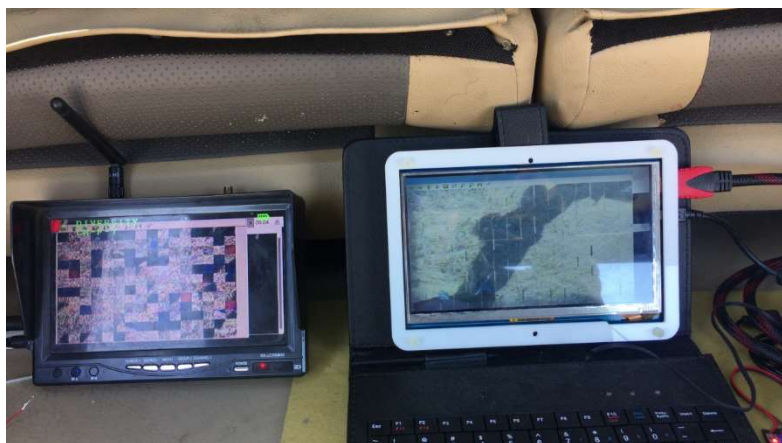
}
return 0;
}

```

LAMPIRAN G

Dokumentasi Kegiatan Tugas Akhir





BIODATA PENULIS



Fery Setiawan lahir di Malang pada tanggal 31 Maret 1990 pernah mengenyam pendidikan Diploma 3 di Politeknik Elektronika Negeri Surabaya. Setelah lulus sempat pula bekerja sebagai *Electronic Technician* di Schlumberger sebuah perusahaan minyak multinasional dan ditempatkan di Balikpapan, Kalimantan Timur selama lebih dari tiga tahun dan akhirnya memutuskan untuk melanjutkan studi pada tahun 2016.

Penulis mempunyai hobi *traveling* dan telah mengunjungi sedikitnya 13 negara baik dalam rangka tugas bekerja maupun *plesiran*. Menjadi mahasiswa setelah lebih dari tiga tahun bekerja tidak semudah yang dibayangkan, namun penulis memiliki tekad kuat untuk melanjutkan studi untuk memperbaiki kualitas hidup dengan memperbaiki pola pikir dengan menempuh pendidikan yang lebih tinggi.

Email :ferykun@gmail.com

Website :feryse.blogspot.com